

ІНФОРМАТИКА, ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА АВТОМАТИЗАЦІЯ

УДК 004.942

DOI <https://doi.org/10.32782/2663-5941/2025.2.2/02>

Барабаш О. В.

Національний технічний університет України
«Київський політехнічний університет імені Ігоря Сікорського»

Свинчук О. В.

Національний технічний університет України
«Київський політехнічний університет імені Ігоря Сікорського»

Шуклін Г. В.

Національний технічний університет України
«Київський політехнічний університет імені Ігоря Сікорського»

Копійка О. В.

Інститут прикладних систем управління Національної академії наук України

ВЕБЗАСТОСУНОК ДЛЯ ДИНАМІЧНОГО ПЕРЕРОЗПОДІЛУ НАВАНТАЖЕННЯ В ІНФОРМАЦІЙНИХ СИСТЕМАХ ЕНЕРГЕТИЧНИХ МЕРЕЖ

У статті розглянуто проблему функціональної стійкості інформаційних систем енергетичних мереж, зокрема питання автоматичного перерозподілу навантаження для забезпечення стабільної та ефективної роботи систем. Запропоновано підхід до балансування навантаження, що ґрунтується на використанні вебзастосунку, який дозволяє динамічно розподіляти задачі між серверами відповідно до їхнього поточного стану та навантаження. Проаналізовано існуючі методи балансування навантаження, включаючи статичні, динамічні та гібридні підходи, а також їхнє застосування в енергетичних інформаційних системах. Визначено основні переваги та недоліки кожного методу з точки зору ефективності, адаптивності та обчислювальних витрат. Описано алгоритми, що реалізовані у вебзастосунку, та підходи до їхнього вибору залежно від особливостей системи та вимог до продуктивності. Розроблений вебзастосунок дозволяє в реальному часі моніторити стан серверів та автоматично здійснювати перерозподіл завдань у разі виявлення перевантаження або відмови окремих вузлів. Запропоноване рішення сприяє підвищенню надійності інформаційних систем, оптимізації використання ресурсів та зниженню ризику збоїв у роботі критичної інфраструктури. Застосування таких підходів у системах енергетичних мереж дозволить не лише підвищити ефективність їхнього функціонування, а й зменшити витрати на експлуатацію за рахунок раціонального розподілу навантаження між серверами. Окрім того, реалізація методів балансування у вигляді вебзастосунку робить можливим інтеграцію запропонованого рішення з існуючими інформаційними системами, що використовуються в енергетичному секторі. Перспективним напрямом подальших досліджень є вдосконалення механізмів прогнозування навантаження, застосування методів машинного навчання для адаптації методів балансування, а також інтеграція розробленого вебзастосунку з хмарними обчислювальними платформами для забезпечення масштабованості системи.

Ключові слова: вебзастосунок, інформаційна система, функціональна стійкість, енергетичні мережі, перерозподіл навантаження, методи балансування, графи, ймовірність.

Постановка проблеми. У сучасних інформаційних системах енергетичних мереж стабільність та безперебійність роботи є критично важливими факторами, особливо під впливом небезпечних

зовнішніх і внутрішніх загроз. Одним із ключових аспектів, що забезпечує надійність таких систем, є функціональна стійкість – здатність системи продовжувати виконувати свої функції в умовах

змінного навантаження, збоїв окремих компонентів або кібератак. Багато наукових робіт присвячено вивченню забезпечення функціональної стійкості інформаційних систем та мереж [1–3]. Розроблені методології забезпечення функціональної стабільності об'єктів критичної інфраструктури [4], запропоновані методи для забезпечення відмовостійкості інформаційної системи та методи для діагностики її несправностей [5–6]. Функціональна стійкість відіграє важливу роль в різних системах: для екологічного моніторингу [7–8], в енергетичних мережах [9], в транспортних мережах [10–11], в промисловій інфраструктурі [12], для об'єктів критичної інфраструктури [13], де висока доступність і надійність є необхідною умовою безперервної роботи, для розробки та вдосконалення веб-сайтів і мобільних додатків [14–15].

Ефективне балансування навантаження запобігає перенавантаженню серверів, підвищує продуктивність і надійність систем. У даній статті буде розглянуто основні принципи балансування навантаження, типи балансувальників та їх вплив на функціональну стійкість вебзастосунку, який буде розроблено для керування енергетичною мережею.

Аналіз останніх досліджень і публікацій. Для забезпечення функціональної стійкості інформаційної системи впровадження ефективних механізмів балансування навантаження між вузлами-серверами дозволяє досягти максимальної ефективності використання ресурсів, знизити час обробки запитів, а також забезпечити стабільність системи у разі збільшення навантаження чи виходу з ладу окремих вузлів.

Існують різні методи для розподілу завдань між вузлами системи [16–17]. Деякі методи пропонують пріоритетність завдань, тоді як інші розподіляють баланс відповідно до довжини завдання. Науковцями всього світу проводяться дослідження щодо існуючих методів перерозподілу навантаження в системі та їх вдосконалення або створення нових [18–19]. Такі дослідження дають іншим дослідникам цінну інформацію про останні досягнення в галузі балансування навантаження для розвитку нових напрямків досліджень [20].

Провідні компанії, як Amazon і Google, успішно використовують балансувальники навантаження для масштабованості та стійкості своїх хмарних сервісів, забезпечуючи безперебійну роботу навіть при високих навантаженнях [21–22].

У сучасних інформаційних системах енергетичних мереж і загалом в ІТ-інфраструктурах

широко використовуються балансувальники навантаження для забезпечення стійкої та ефективної роботи систем. Розглянемо найпопулярніші з них для встановлення їхніх переваг та недоліків, на основі яких можна буде спроектувати власну систему.

NGINX – продуктивний балансувальник навантаження з підтримкою HTTP(S) і TCP/UDP. Забезпечує гнучкість, інтеграцію з Docker і Kubernetes, але потребує додаткових інструментів для моніторингу [23].

HAProxy – потужний балансувальник навантаження з підтримкою HTTP(S) і TCP. Забезпечує безпеку, стабільність і контроль трафіку, але вимагає ручного налаштування, що ускладнює роботу для початківців. Крім того, динамічне управління трафіком потребує додаткових інструментів [24].

F5 BIG-IP – потужне рішення для балансування навантаження з глибоким контролем трафіку та розширеною безпекою. Підтримує API-автоматизацію та інтеграцію, але має високу вартість і складне налаштування [25].

Google Cloud Load Balancer – хмарне рішення для глобального балансування трафіку з автоматичним масштабуванням. Забезпечує високу продуктивність, але залежить від Google Cloud і може бути складним у налаштуванні [22].

Проаналізувавши популярні наявні рішення, було виявлено, що існуючі рішення ефективні, але дорогі, складні в налаштуванні та потребують додаткових інструментів. Дослідження не враховують специфіку архітектури систем і не дають чітких рекомендацій щодо методів оцінки.

Постановка завдання. Метою статті є розробка програмного забезпечення для ефективного перерозподілу навантаження в інформаційних системах енергетичних мереж на основі сучасних методів балансування.

Виклад основного матеріалу. Методи перерозподілу навантаження можна поділити за інформацією про стан системи [26]:

- статичні – використовують фіксовані правила, незалежні від поточного стану системи. Вони вимагають попередніх даних про ресурси кожного вузла та не підходять для динамічних систем. Оптимальна робота можлива лише за незначних коливань навантаження. Хоча такі алгоритми прості у використанні, вони часто не враховують підключені сервери, що спричиняє дисбаланс ресурсів. Отже, вони ефективні лише для стабільних систем, але не адаптуються до змін;

- динамічні – приймають рішення в реальному часі, враховуючи поточний стан системи.

Вони дозволяють передавати завдання з перевантажених вузлів на менш завантажені, регулярно оцінюючи робоче навантаження. Система обмінюється даними про стан вузлів і перерозподіляє ресурси під час виконання завдань. Ці методи більш адаптивні, ніж статичні, але потребують додаткових обчислювальних ресурсів;

- гібридні методи – поєднують переваги статичних і динамічних алгоритмів, оптимізуючи управління вузлами та підвищуючи ефективність. При перевантаженні вузол перенаправляє нове завдання на менш завантажений. Водночас ці методи мають затримки у передачі та плануванні, що потребує вдосконалення.

Отже, ефективне балансування навантаження може мінімізувати використання ресурсів, що призведе до додаткової економії споживання енергії. Розглянемо класифікацію статистичних та динамічних методів перерозподілу навантаження.

Класифікація статистичних балансувальників:

- 1) за методом розподілу навантаження – round-robin (черговий принцип), розподіл за IP-адресою, розподіл за типом задач;

- 2) за типом ресурсу, що розподіляється – обчислювальні та мережеві ресурси;

- 3) за архітектурою системи – централізовані та розподілені балансувальники.

Класифікація динамічних балансувальників:

- 1) за методом балансування – розподіл на основі стратегії, пріоритетне навантаження;

- 2) за рівнем інтеграції – внутрішні та зовнішні балансувальники;

- 3) за способом моніторингу – активні та пасивні балансувальники.

Отже, статистичні та динамічні балансувальники мають широкий спектр використання під свої специфічні потреби. У таблиці 1 наведено ключові відмінності статичних та динамічних балансувальників навантаження.

Отже, статистичні та динамічні балансувальники мають свої характеристики та переваги, зокрема у контексті енергетичних мереж. Це дозволяє створити надійнішу, гнучкішу та про-

дуктивнішу інфраструктуру, здатну впоратися з різноманітними викликами, які ставлять перед нею умови сучасного ринку.

Під час розроблення власного вебзастосунку були обрані і статистичні, і динамічні методи балансування.

Статистичні балансувальники будуть використовуватися:

- на початкових етапах розподілу задач між вузлами, створюючи базову систему управління навантаженням для рівномірного розподілу ресурсів;

- при стабільному або передбачуваному навантаженні, оскільки вони не адаптуються до змін у реальному часі. Наприклад, якщо вузли відповідають за конкретні типи завдань, статичний підхід знижує складність системи та забезпечує стабільність. Проте у разі різкого зростання навантаження або збою ефективність може знижуватися;

- для рівномірного розподілу запитів у енергетичних системах, коли кожен тип запиту спрямовується на сервер, що спеціалізується на його обробці;

- для підвищення продуктивності та надійності, розподіляючи ресурси відповідно до потужності серверів. Завдання з вищими вимогами спрямовуються на потужні вузли, а простіші – на менш потужні, що забезпечує ефективне використання ресурсів.

Динамічні балансувальники будуть використовуватися:

- для ефективного розподілення навантаження між вузлами, забезпечуючи стабільну роботу системи навіть у випадках різких коливань навантаження;

- для швидкого реагування на збої в роботі окремих серверів, перенаправляючи запити на доступні ресурси, що значно підвищує надійність системи.

Даний вибір балансувальників зумовлений необхідністю адаптації до змін навантаження в енергомережах. Динамічний балансувальник

Таблиця 1

Порівняння балансувальників навантаження

Параметри	Статистичне балансування	Динамічне балансування
Коливання вхідного навантаження у системі	Низьке	Високе
Розподіл трафіку між серверами	Рівномірний	Динамічний
Глибокі знання про доступні ресурси системи	Вимагаються	Не обов'язкові
Потреба комунікації з серверами у реальному часі	Наявна	Відсутня
Можливість перерозподілу навантаження у реальному часі	Відсутня	Наявна для зменшення недостатньо навантажених серверів

оперативно оптимізує ресурси, запобігаючи перевантаженню серверів, тоді як статистичний підходить для обмежених ресурсів. Гнучка зміна методів розподілу підвищує надійність і адаптивність системи.

Були обрані такі методи перерозподілу: статистичні (Round robin, Weighted round robin connection, IP hash, Domain hash, Randomizer connection) та динамічні (Weighted least connection, Least connection, Weighted response time, Resource based, Shortest Job Next).

Для побудови програмного забезпечення було використано клієнт-серверну архітектуру. Це система, що зберігає, контролює та надає ресурси і сервіси за запитом клієнта. У такій системі будь-яка взаємодія здійснюється за допомогою мережі. На рисунку 1 продемонстровано схему такої архітектури.

Клієнтом може бути будь-який пристрій збору даних (датчики на підстанціях, лініях електропередач та інших елементах енергетичної мережі, системи моніторингу та діагностики, лічильники) або програмне забезпечення, яке надсилає запити на сервер для отримання даних, їх збереження, виконання обчислень тощо. Зазвичай клієнт реалізований у вигляді браузера або мобільного застосунку. Сервером може бути центральний сервер управління, сервери баз даних, сервери додатків, тобто пристрій, який приймає запити від клієнта, обробляє їх і надає відповідь. Клієнт і сервер можуть бути представлені як одним пристроєм, так і різними, що взаємодіють через мережеві протоколи, такі як HTTP, FTP або SMTP. Така архітектура забезпечує ефективний розпо-

діл ресурсів, централізовану обробку інформації та масштабування застосунку для обслуговування різної кількості клієнтів.

Було розроблено веб-застосунок для моніторингу та управління системою вузлів-серверів, що виконують завдання, отримані через REST-запити. Завдання автоматично розподілятимуться між вузлами за допомогою балансувальника навантаження.

Основні функції вебзастосунку:

- моніторинг стану вузлів: відображення навантаження, часу роботи, використання ресурсів та доступності;
- редагування конфігурації вузлів: додавання, видалення та налаштування параметрів вузлів;
- розподіл завдань: автоматичний розподіл завдань між вузлами з урахуванням пріоритетів та навантаження;
- перерозподіл завдань: автоматичний перерозподіл завдань у разі виходу вузла з ладу.

Застосунок забезпечує функціональну стійкість інформаційної системи енергетичної мережі, дозволяючи оперативно реагувати на зміни та дестабілізуючі фактори. Передбачена інтеграція із зовнішніми системами для збору та аналізу даних у реальному часі.

Для ефективної роботи системи розподілу навантаження необхідна наявність доступу до мережі Інтернет на пристрої або сервері, де розміщено застосунок. Оскільки система повинна оперувати актуальними даними, а нові завдання надходять з різних джерел (наприклад, з місцевої підстанції) через Інтернет, стабільне з'єднання є критично важливим як для відправника, так і для

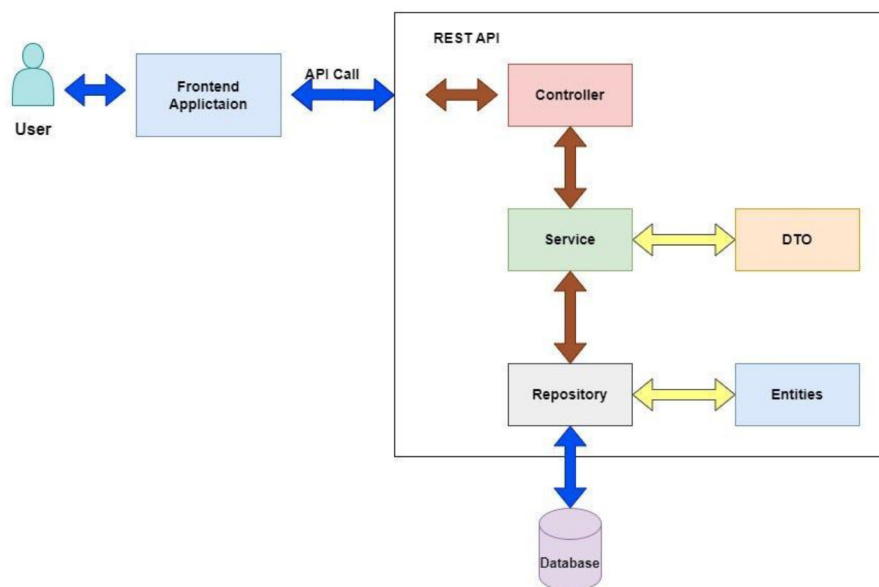


Рис. 1. Архітектура застосунку

отримувача. Крім того, всі відправники повинні бути ознайомлені з API системи, тобто необхідно чітко визначити, що і в якому форматі слід передавати до фінальної системи, яка відповідає за розподіл навантаження між своїми вузлами.

Після входу на сайт користувач потрапляє на головну сторінку з навігаційним меню та лаконічним описом функціоналу. Він переходить до розділу «Мережі» за допомогою навігації. Тут користувач може:

- створити нову мережу, задавши їй унікальну назву та обравши метод розподілу навантаження;
- переглянути список вже зареєстрованих мереж;
- відредагувати назву або метод розподілу навантаження для будь-якої з існуючих мереж;
- видалити мережу.

Про кожну мережу можна отримати детальну інформацію: мінімальне, максимальне та середнє навантаження, сумарний час виконання задач та скільки їх вже було виконано, статистика функціонуючих вузлів (рис. 2).

Користувач може переглянути стан системи у вигляді графового представлення (рис. 3). Граф

відображає актуальну інформацію про кожну мережу: вузли (трикутники) та задачі (квадрати), присвоєні до них. Мережі представлені колами. Користувач може переглядати поточний стан мереж, змінювати положення елементів та оновлювати сторінку для відображення змін.

У розділі «Вузли» користувач може створювати нові вузли для будь-якої з мереж, представлених у системі. Для створення вузла необхідно вибрати бажану мережу, задати початковий стан, вказати місткість, вагу, ймовірність виходу з ладу та порт. Усі створені вузли відображаються в меню «Список». Кожен вузол у списку має фон, колір якого відповідає його поточному стану. Користувач також має можливість вручну змінювати стан та інші характеристики вузла, або видалити його з системи. Візуальне представлення сторінки наведено на рисунку 4.

У розділі «Методи» користувач може зареєструвати метод, який має бути попередньо реалізований у застосунку (рис. 5). Важливо, щоб усі алгоритми мали однакову структуру та були правильно оголошені, щоб на них можна було посилатися через інтерфейс. Для цього потрібно

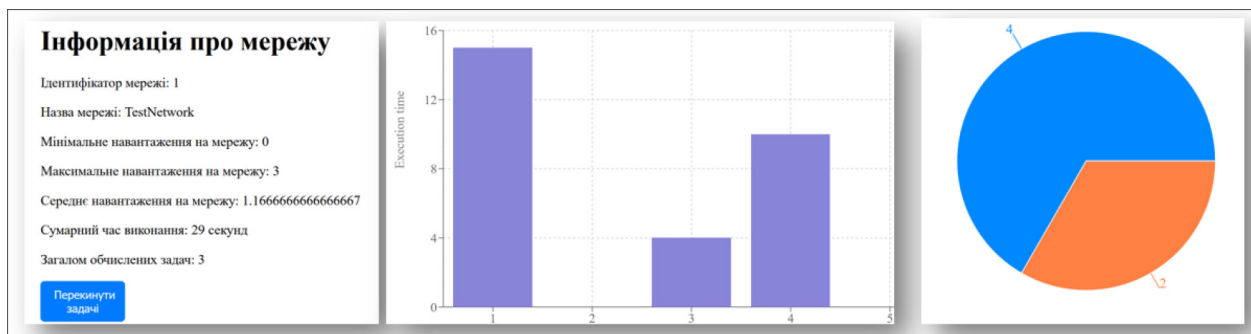


Рис. 2. Інформація про мережу

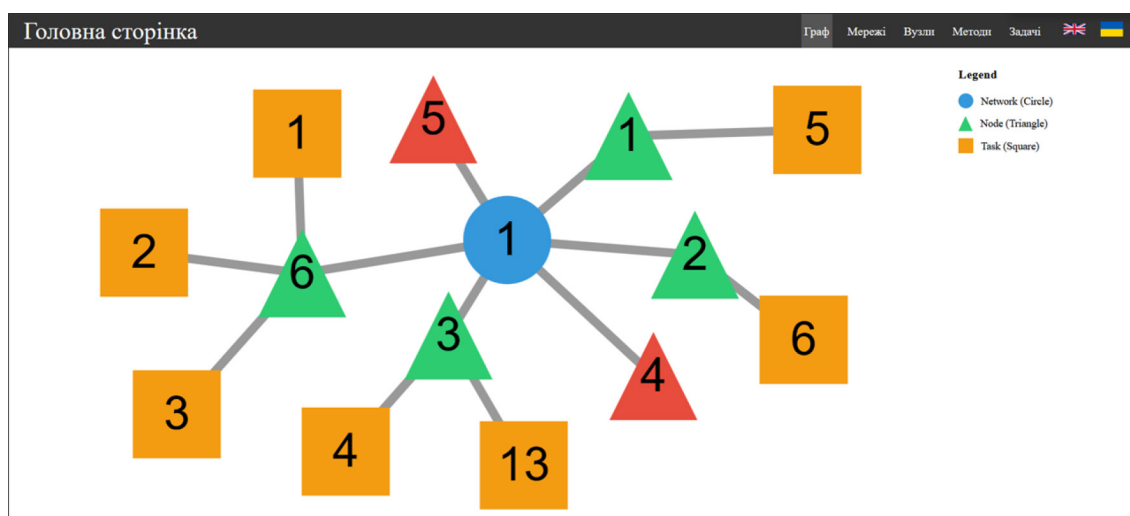


Рис. 3. Графове представлення системи

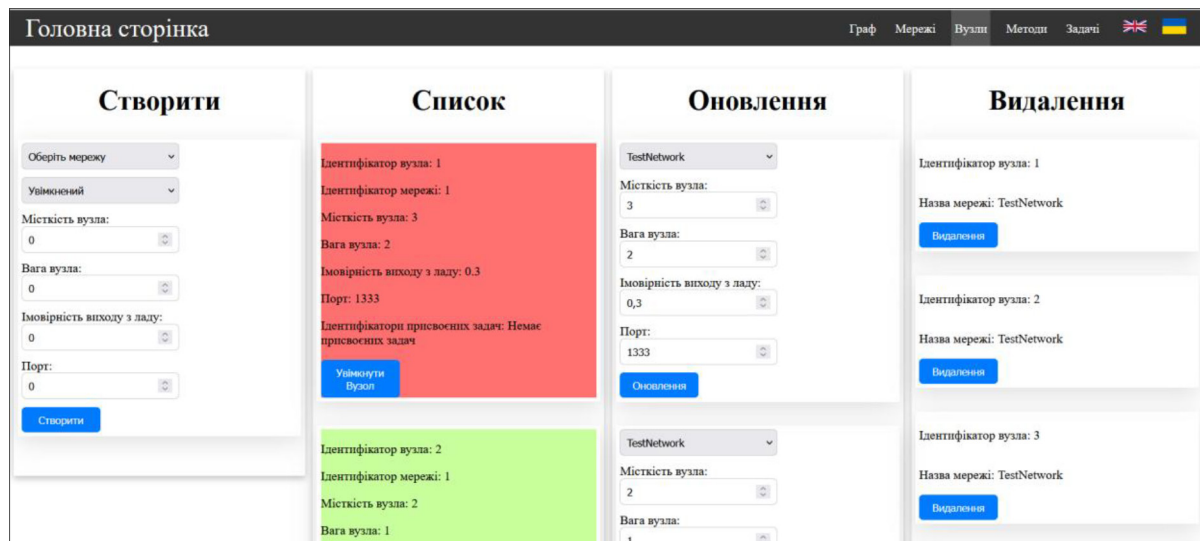


Рис. 4. Керування вузлами

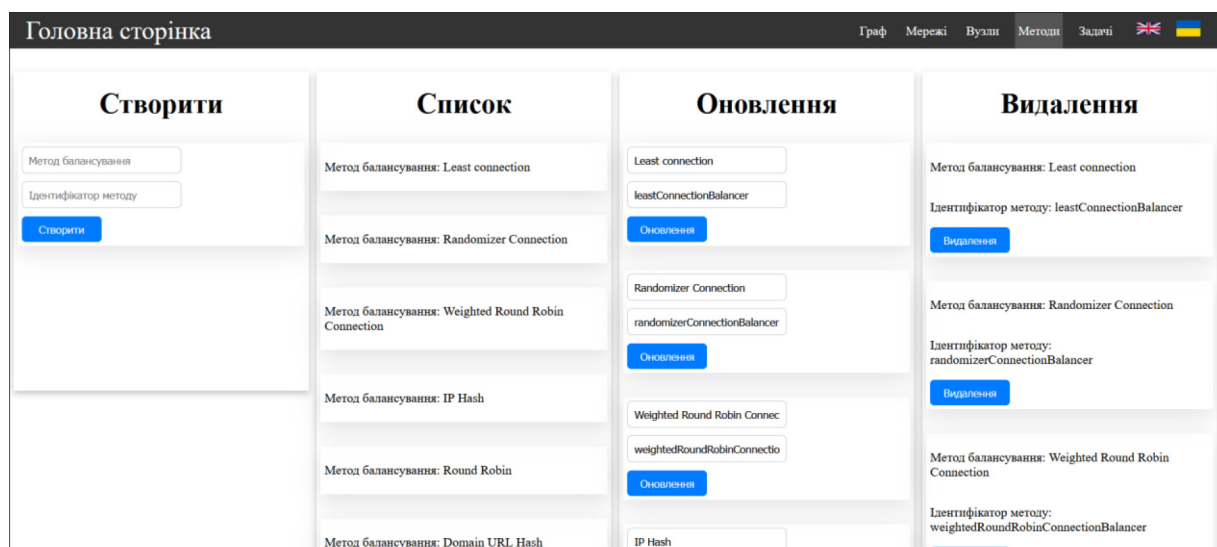


Рис. 5. Налаштування методів

вказати назву, яку розробник дав відповідному алгоритму, як ідентифікатор. Якщо назви не збігаються, перерозподіл не відбудеться. Аналогічно, на цій сторінці можна переглядати, редагувати та видаляти вже зареєстровані методи.

У розділі «Завдання» користувач може емулявати завдання з зовнішніх джерел. Тут можна реєструвати нові задачі, переглядати існуючі та видаляти їх. На відміну від інших розділів, тут неможливо оновити задачу. Це пов'язано з тим, що після призначення задачі вузлу починається її виконання та облік часу. Зміна будь-якого поля задачі потребуватиме її перезапуску, що призведе до неоднозначності та негативно вплине на процеси системи, що залежать від поточних даних. У реальних умовах ці дані можуть заповнюватися автоматично пристроєм-датчиком, який збирає

інформацію та надсилає її на сервер для перерозподілу.

У результаті розробки було створено застосунок, який дозволяє аналізувати та тестувати обрані методи розподілу навантаження, моделювати вихід з ладу частини системи та перевіряти перерозподіл завдань у подібних ситуаціях. Створений застосунок надає користувачеві можливість створювати та налаштовувати мережі розподілу навантаження відповідно до поставлених завдань. Реалізовані функції перерозподілу навантаження нових завдань та завдань вузла, що вийшов з ладу, дають змогу аналізувати та визначати ефективність обраних алгоритмів.

Отже, розроблений застосунок є ефективним інструментом для аналізу, тестування та конфігурування мереж розподілу навантаження, а також

для моделювання та дослідження сценаріїв відмови системи.

Висновки. У статті досліджено проблему стійкості інформаційних систем енергомереж через ефективне балансування навантаження. Проаналізовано методи розподілу, їхні переваги й недоліки, а також обґрунтовано вибір підходів для вебзастосунку, що автоматично керує навантаженням. Розроблений вебзастосунок адаптивно перерозподіляє навантаження в реальному часі,

зменшуючи ризик перенавантаження серверів та забезпечуючи безперебійну роботу. Поєднання статичних і динамічних методів дозволяє обирати алгоритм залежно від умов системи.

Запропоноване рішення підходить для систем різної складності, підвищуючи стабільність критичної інфраструктури. Подальші дослідження можуть зосередитися на вдосконаленні адаптивного балансування, інтеграції з машинним навчанням і прогнозуванні навантаження.

Список літератури:

1. Mashkov O., Kosenko V. Ensuring of functional stability of difficult dynamic systems as one of urgent scientific tasks of modern theory of automatic control. *Informatyka, Automatyka, Pomiar w Gospodarce i Ochronie Środowiska*. 2015. Vol. 5, no. 3. P. 39–42. <https://doi.org/10.5604/20830157.1166550>
2. Mashkov O., Chumakevych V., Sokulsky O., Chyrun L. Features of determining controlling effects in functionally-stable systems with the recovery of a control. *Mathematical Modeling And Computing*. 2019. Vol. 6, no. 1. P. 85–91. <http://doi.org/10.23939/mmc2019.01.085>
3. Mashkov O.A., Chumakevich V.A., Mamchur Yu.V., Kosenko V.R. The method of inverse problems of dynamics for the synthesis of a system of stabilization of the movement of a dynamic object on operatively programmable trajectories. *Mathematical Modeling And Computing*. 2020. Vol. 7, no. 1. P. 29–38. <http://doi.org/10.23939/mmc2020.01.029>
4. Barabash O., Sobchuk V., Musienko A., Laptiev O., Bohomia V., Kopytko S. System Analysis and Method of Ensuring Functional Sustainability of the Information System of a Critical Infrastructure Object. In: Zgurovsky, M., Pankratova, N. (eds.). *System Analysis and Artificial Intelligence. Studies in Computational Intelligence*. 2023. Vol. 1107. Springer, Cham. P. 117–192. https://doi.org/10.1007/978-3-031-37450-0_11
5. Peng S.-L., Lin C.-K., Tan J.J.M. and Hsu L.-H. The g-Good-Neighbor Conditional Diagnosability of Hypercube under PMC Model. *Applied Mathematics and Computation*. 2012. Vol. 218, no. 21. P. 10406–10412. <https://doi.org/10.1016/j.amc.2012.03.092>
6. Yuan J., Liu A., Ma X., Liu X., Qin X. and Zhang J. The g-Good-Neighbor Conditional Diagnosability of k-Ary n-Cubes under the PMC Model and MM Model. *IEEE Transactions on Parallel and Distributed Systems*. 2015. Vol. 2, no. 4. P. 1165–1177. <http://doi.org/10.1109/TPDS.2014.2318305>
7. Barabash O., Svynchuk O., Bandurka O., Ilin O. Information system for detection of parameters of dangerous industrial facilities based on geoinformation technologies. *Informatyka, Automatyka, Pomiar W Gospodarce i Ochronie Środowiska*. 2024. Vol. 14, no. 3. P. 9–14. <https://doi.org/10.35784/iapgos.6093>
8. Карпенко М.І., Чумаченко С.М., Мошенський А.О., Гуйда О.Г. Аргументація ефективності альтернативних засобів моніторингу на прикладі Meshtastic та APRS. Вчені записки ТНУ імені В.І.Вернадського. Серія: Технічні науки, 2024, Том 35 (74), № 5. С. 154–160. <https://doi.org/10.32782/2663-5941/2024.5.1/24>
9. Barabash O., Svynchuk O., Salanda I., Mashkov V., Myroniuk M. Ensuring the Functional Stability of the Information System of the Power Plant on the Basis of Monitoring the Parameters of the Working Condition of Computer Devices. *Advanced Information Systems*. 2024. Vol. 8, no. 2. P. 107–117. <https://doi.org/10.20998/2522-9052.2024.2.12>
10. Бандурка О., Романів Р., Свинчук О. Архітектура розподіленої функціонально стійкої системи моніторингу транспортних засобів на базі блокчейн-технології. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2025, № 1. С. 86–96. <https://doi.org/10.31891/2219-9365-2025-81-11>
11. Кисельов В.Б., Лановий О.Т., Кошарний О.М. Математичне моделювання впливу функціонування мережі автомобільних доріг на економіку України. Вчені записки ТНУ імені В.І.Вернадського. Серія: Технічні науки, 2022, Том 33(72), № 1. С. 300–307. <https://doi.org/10.32838/2663-5941/2022.1/46>
12. Собчук В.В., Мусієнко А.П., Ільїн О.Ю. Аналіз використання ієрархічної структури для забезпечення функціональної стійкості автоматизованої системи управління підприємством. *Телекомунікаційні та інформаційні технології*. 2018. № 4(61). С. 53–61. <http://doi.org/10.31673/2412-4338.2018.045361>
13. Laptiev O., Barabash O., Tsyganivska I., Obidin D., Sobchuk A. The method of construction of the law of safety management of critical infrastructure objects under the conditions of external uncontrolled influences. *CEUR Workshop Proceedings*. 2023. Vol. 3624. P. 291–300. https://ceur-ws.org/Vol-3624/Paper_24.pdf
14. Дроменко В.Б., Пилипенко Ю.М., Кисельов В.Б. Застосування рекурсивних алгоритмів і функцій по декількох змінних при організації циклічних процесів. Вчені записки ТНУ імені В.І.Вернадського. Серія: Технічні науки, 2023, Том 34(73), № 3. С. 136–140. <https://doi.org/10.32782/2663-5941/2023.3.1/21>

15. Василенко В.М., Карпенко М.І., Скибінський А.С., Гуйда О.Г. Аналіз переваг та обмежень Low-Code платформ на прикладі Webflow.com. Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки, 2024, Том 35(74), № 2. С. 65–70. <https://doi.org/10.32782/2663-5941/2024.2/09>
16. Shafiq D.A., Jhanjhi N., Abdullah A. Load balancing techniques in cloud computing environment: a review. *Journal of King Saud University – Computer and Information Sciences*. 2022. Vol. 34, no. 7. P. 3910–3933. <https://doi.org/10.1016/j.jksuci.2021.02.007>
17. Барабаш О.В., Свинчук О.В., Шуклін Г.В. Алгоритм перерозподілу навантаження для забезпечення функціональної стійкості розподілених вебзастосунків. *Зв'язок*. 2024. № 5. С. 3–11. <https://doi.org/10.31673/2412-9070.2024.050183>
18. Gupta Y. Novel distributed load balancing algorithms in cloud storage. *Expert Systems with Applications*. 2021. Vol. 186. 115713. <https://doi.org/10.1016/j.eswa.2021.115713>
19. Velpula P, Pamula R. EBGO: an optimal load balancing algorithm, a solution for existing tribulation to balance the load efficiently on cloud servers. *Multimedia Tools and Applications*. 2022. Vol. 81, no. 24. P. 34653–34675. <https://doi.org/10.1007/s11042-021-11012-w>
20. Ahmed M.K., Salman S.A., Abdulhammed O.Y. Load balancing techniques in cloud computing: A review. *Samarra Journal of Pure and Applied Science*. 2024. Vol. 6, no. 1. P. 223–250. <https://doi.org/10.54153/sjpas.2024.v6i1.526>
21. AWS Elastic Load Balancing Documentation. Amazon Web Services. URL: <https://surl.lu/cwxdzs> (дата звернення: 11.03.2025).
22. Google Cloud: Load Balancing Overview. Google Cloud Documentation. URL: <https://cloud.google.com/load-balancing> (дата звернення: 11.03.2025).
23. Nginx Load Balancing Documentation. Nginx. URL: <https://surl.li/naqlzs> (дата звернення: 11.03.2025).
24. HAProxy config tutorials. URL: <https://surl.li/ckcvaf> (дата звернення: 11.03.2025).
25. BIG-IP Administrator Guide v4.0: Inbound Load Balancing. F5 Networks. URL: <https://surl.li/qfemrm> (дата звернення: 11.03.2025).
26. Chen S.L., Chen Y.Y., Kuo S.H. CLB: A novel load balancing architecture and algorithm for cloud services. *Computers & Electrical Engineering*. 2017. Vol. 58. P. 154–160. <https://doi.org/10.54153/sjpas.2024.v6i1.526>

Barabash O.V., Svyinchuk O.V., Shuklin G.V., Koptika O.V. WEB APPLICATION FOR DYNAMIC LOAD REDISTRIBUTION IN INFORMATION SYSTEMS OF ENERGY NETWORKS

This paper addresses the issue of functional resilience in information systems for energy networks, specifically focusing on the problem of automatic load redistribution to ensure stable and efficient system operation. As energy networks become increasingly complex, ensuring reliable data processing and task execution requires adaptive and intelligent load-balancing mechanisms. To tackle this challenge, a load-balancing approach is proposed, based on the use of a web application that dynamically distributes tasks among servers according to their current state, workload, and performance metrics. Existing load-balancing methods, including static, dynamic, and hybrid approaches, are analyzed, along with their application in energy information systems. The advantages and limitations of each method are examined in terms of efficiency, adaptability, fault tolerance, and computational overhead. The study highlights the necessity of real-time load management to prevent bottlenecks, optimize resource utilization, and ensure continuous system availability. The developed web application implements intelligent load-balancing algorithms that enable real-time server monitoring and automatic task redistribution in cases of server overload or failure. These algorithms consider multiple factors such as server capacity, response time, and network conditions to optimize load distribution dynamically. The proposed solution not only enhances system reliability but also minimizes latency and improves overall system performance. Moreover, the implementation of load-balancing techniques as a web application facilitates seamless integration with existing information systems in the energy sector. This integration helps optimize resource allocation, reduce operational and maintenance costs, and improve decision-making processes in energy management. A promising direction for future research includes refining load forecasting models, leveraging machine learning techniques to enhance load-balancing strategies, and integrating the developed web application with cloud computing platforms to ensure high scalability and fault tolerance.

Key words: web application, information system, functional stability, energy networks, load redistribution, load-balancing methods, graphs, probability.