

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія

на тему «Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів»

КвРКІП. 2301154.23.01.12 ПЗ

Виконав: студент 2 курсу, група КІ2м-23-1


Підпис

Павло КАРВАЦЬКИЙ
Ім'я, прізвище

Керівник д-р. техн. наук, професор
Науковий ступінь, вчене звання


Підпис

Олег САВЕНКО
Ім'я, прізвище

До захисту допускаю:

Зав. кафедри КІС, доктор філософії, доцент

Ольга ПАВЛОВА 

02 05 2025 р.

Хмельницький, 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-НАУКОВА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПІВЛОВА



“ 01 ” 09 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Павлу КАРВАЦЬКОМУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів

Керівник проекту (роботи) Олег САВЕНКО д.т.н., професор

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.01.2025 №8

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Основи керування потужністю в комп'ютерних мережах

Аналіз існуючих підходів до керування потужністю системи в комп'ютерній мережі

Система динамічного керування живленням

Оцінка ефективності методу керування потужністю в комп'ютерній системі

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сергій ЛИСЕНКО, професор кафедри КПС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КПС		



7. Дата видачі завдання « 01 » 09 2024р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики з керівником	01.09.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження;	01.10.2024	виконано
3	Робота над розділом 1 –аналіз існуючих підходів до керування потужністю системи	01.11.2024	виконано
4	Робота над розділом 2 – основи керування потужністю в комп'ютерних мережах	01.12.2024	виконано
5	Робота над науковою статтею	01.02.2025	виконано
6	Робота над розділом 3 – система динамічного керування живленням	15.02.2025	виконано
7	Робота над розділом 4 – оцінка ефективності методу керування потужністю в комп'ютерній системі	01.04.2025	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2025	виконано
9	Попередній захист ДРМ	29.04.2025	виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2025	

Студент

Керівник роботи


Підпис

Підпис

Павло КАРВАЦЬКИЙ

Ім'я, прізвище

Олег САВЕНКО

Ім'я, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів

Автор роботи: Павло КАРВАЦЬКИЙ

Керівник роботи: Олег САВЕНКО

Пояснювальна записка: 73 с., 19 рис., 6 табл., 3 дод., 80 джерел.

КЕРУВАННЯ ПОТУЖНІСТЮ, КОМП'ЮТЕРНА МЕРЕЖА, ПРОФІЛІ РОБОЧИХ ПАРАМЕТРІВ, ОПТИМІЗАЦІЯ ЕНЕРГОСПОЖИВАННЯ, АДАПТИВНЕ КЕРУВАННЯ, ЕНЕРГОЕФЕКТИВНІСТЬ, СИСТЕМИ РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ, МЕРЕЖЕВІ РЕСУРСИ, ІНТЕЛЕКТУАЛЬНІ АЛГОРИТМИ, МОДЕЛЮВАННЯ НАВАНТАЖЕННЯ.

Об'єктом дослідження є процеси керування потужністю в комп'ютерних мережах.

Предметом дослідження є методи та алгоритми керування потужністю комп'ютерної мережі на основі аналізу профілів робочих параметрів.

Метою кваліфікаційної роботи магістра є покращення керування потужністю в комп'ютерній мережі відповідно до динамічних профілів навантаження та робочих параметрів з метою підвищення енергоефективності та стабільності роботи системи.

Для розв'язання поставлених задач використовувалися методи:

- системного аналізу;
- математичного моделювання;
- теорії графів та мереж;
- алгоритмів оптимізації;
- комп'ютерного моделювання та програмної реалізації.

Наукова новизна отриманих результатів: розроблено новий метод керування потужністю, який враховує адаптивну зміну профілів навантаження та робочих параметрів у комп'ютерній мережі;

На основі проведених досліджень розроблено метод керування потужністю з урахуванням робочих параметрів в комп'ютерній мережі.

Практична значимість отриманих результатів полягає у можливості впровадження розробленого методу та програмного забезпечення в сучасні розподілені мережеві системи для оптимізації споживання електроенергії, підвищення ефективності використання обчислювальних ресурсів та забезпечення стійкості до пікових навантажень.

У вступі подано об'єкт та предмет дослідження, мету, наукову новизну та практичну цінність роботи, а також характеристику структури роботи.

У першому розділі проведено аналіз відомих рішень щодо керування навантаження в мультимедійній системі.

У другому розділі здійснено визначення основ керування потужністю в комп'ютерних мережах за різними робочими параметрами.

У третьому розділі розроблено метод динамічного керування потужністю з урахуванням змінних параметрів мережі.

У четвертому розділі здійснено моделювання та експериментальне оцінювання ефективності запропонованого методу.

У висновках підведено підсумки досягнення результатів з розв'язання завдань дослідження.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	5
ВСТУП	6
1 АНАЛІЗ ВІДОМИХ МЕТОДІВ ТА ЗАСОБІВ КЕРУВАННЯ ПОТУЖНІСТЮ В КОМП'ЮТЕРНИХ МЕРЕЖАХ	8
1.1 Поняття енергоспоживання в комп'ютерних мережах	8
1.2 Огляд сучасних методів оптимізації енергоспоживання	13
1.3 Постановка задачі	20
1.4 Висновки	21
2 ОСНОВИ ДО КЕРУВАННЯ ПОТУЖНІСТЮ В КОМП'ЮТЕРНИХ МЕРЕЖАХ ЗА РІЗНИМИ РОБОЧИМИ ПАРАМЕТРАМИ	22
2.1 Методи аналізу енергоспоживання вбудованих систем на ранніх етапах проектування	22
2.2 Метод оцінки енергії на системному рівні	23
2.3 Архітектура розробленого інструменту	27
2.4 Висновки	31
3 МЕТОД КЕРУВАННЯ ПОТУЖНІСТЮ З УРАХУВАННЯМ ЗМІННИХ ПАРАМЕТРІВ МЕРЕЖІ	32
3.1 Оптимізація політик динамічного керування живленням (DPM)	32
3.2 Виклики прогнозування навантаження та переходу між станами живлення системи	41
3.3 Метод керування потужністю з урахуванням змінних параметрів мережі	47
3.4 Висновки	55
4 ОЦІНКА ЕФЕКТИВНОСТІ МЕТОДУ КЕРУВАННЯ ПОТУЖНІСТЮ В КОМП'ЮТЕРНІЙ СИСТЕМІ	58

	4
4.1 Визначення основних параметрів та дослідження характеристик	58
4.2 Архітектура програмної реалізації інформаційної технології	60
4.3 Експериментальні дослідження	63
4.4 Висновки	76
ВИСНОВКИ	77
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	79
ДОДАТОК А Лістинг програмного забезпечення	86
ДОДАТОК Б Наукова праця здобувача	89
ДОДАТОК В Презентація	94

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AWS - Amazon Web Services

CPU - центральний процесор

GPU - графічний процесор

DVFS - Dynamic Voltage and Frequency Scaling

NIC - мережевий інтерфейсний адаптер

KVM - система віртуалізації для Linux (Kernel-based Virtual Machine)

QoS - Quality of Service (якість обслуговування)

CDN - система розподілених серверів (Content Delivery Network)

API - інтерфейс для взаємодії між програмами

SCADA - система моніторингу та керування технологічними

PLC - програмований логічний контролер (Programmable Logic Controller)

DRS - компонент VMware (Distributed Resource Scheduler)

GRP - Green Routing Protocol (протокол маршрутизації для оптимізації)

AI - штучний інтелект (Artificial Intelligence)

ВСТУП

В умовах стрімкого розвитку інформаційних технологій та зростаючого навантаження на комп'ютерні мережі питання енергоефективності та оптимізації потужності стає надзвичайно актуальним. Використання сучасних методів керування потужністю в мережевих системах дозволяє не лише зменшити енергоспоживання, але й підвищити загальну продуктивність інфраструктури, забезпечуючи стабільність її роботи.

Зі збільшенням кількості мережевих пристроїв та обсягів передавання даних особливої важливості набуває адаптивне регулювання енергоспоживання відповідно до змінних профілів робочих параметрів. Використання таких методів дозволяє мінімізувати втрати енергії без шкоди для продуктивності мережевих сервісів, що є важливим аспектом у сучасних системах зв'язку, центрах обробки даних та хмарних середовищах.

Актуальність дослідження зумовлена потребою у створенні гнучких механізмів управління потужністю, які можуть адаптуватися до змінних умов роботи мережі, враховуючи параметри навантаження, рівень використання ресурсів та характеристики окремих компонентів інфраструктури. Такий підхід є ефективним як для великих корпоративних мереж, так і для розподілених хмарних платформ, де оптимізація енергоспоживання сприяє зниженню експлуатаційних витрат.

Метою кваліфікаційної роботи магістра є покращення керування потужністю в комп'ютерній мережі відповідно до динамічних профілів навантаження та робочих параметрів з метою підвищення енергоефективності та стабільності роботи системи.

Для досягнення поставленої мети необхідно вирішити такі основні завдання:

- проаналізувати існуючі підходи до керування потужністю в мережевих системах;

- дослідити вплив основних профілів робочих параметрів на ефективність керування потужності;
- розробити метод динамічного керування потужністю з урахуванням змінних параметрів мережі;
- провести моделювання та експериментальне оцінювання ефективності запропонованого методу.

Об'єктом дослідження є процеси керування енергоспоживанням у комп'ютерних мережах, а предметом – методи оптимізації потужності системи на основі адаптивного аналізу робочих параметрів.

Предметом дослідження є методи та алгоритми керування потужністю комп'ютерної мережі на основі аналізу профілів робочих параметрів.

Наукова новизна отриманих результатів: розроблено новий метод керування потужністю, який враховує адаптивну зміну профілів навантаження та робочих параметрів у комп'ютерній мережі.

На основі проведених досліджень розроблено метод керування потужністю з урахуванням робочих параметрів в комп'ютерній мережі.

Практичне значення результатів полягає у можливості впровадження розробленого методу у корпоративні мережі, центри обробки даних та телекомунікаційні системи для підвищення ефективності використання енергоресурсів.

Для розв'язання поставлених задач використовувались методи керування, оптимізація навантаження в комп'ютерній мережі.

За темою кваліфікаційної роботи опубліковано одну статтю Клейн О.М., Карвацький П.А. Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів

1 АНАЛІЗ ВІДОМИХ МЕТОДІВ ТА ЗАСОБІВ КЕРУВАННЯ ПОТУЖНІСТЮ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

1.1 Поняття енергоспоживання в комп'ютерних мережах

Енергоспоживання в комп'ютерних мережах є критично важливим аспектом, що впливає на ефективність їх функціонування та загальні експлуатаційні витрати. Воно характеризується кількістю енергії, споживаної пристроями та вузлами мережі, такими як маршрутизатори, комутатори, сервери, робочі станції тощо [1]. У сучасних умовах швидкого розвитку цифрових технологій та зростання обсягів обробки даних оптимізація енергоспоживання стає дедалі актуальнішою .

Для аналізу енергоспоживання враховують такі параметри:

1. Робоче навантаження: кількість переданих та оброблених даних.
2. Топологія мережі: структура з'єднань між вузлами.
3. Типи обладнання: характеристики пристроїв, їх ефективність.
4. Режими роботи: активність компонентів у певний момент часу [2, 3].
5. Тип обладнання: сервери, маршрутизатори, комутатори, робочі станції.
6. Характеристики мережі: топологія, обсяг переданих даних, середня затримка.
7. Режими роботи обладнання: активний, режим очікування і "сплячий режим" [4].

Реальними прикладами є те, що у дата-центрах великих корпорацій, таких як Google та Amazon, на електроенергію витрачається до 50% загального бюджету. У побутових умовах енергоспоживання роутерів та мережевих пристроїв може складати до 10% від загального споживання електроенергії домогосподарства [5].

На рис. 1.1 показано діаграму розподілу енергоспоживання за типами пристроїв у типовій мережі.

Додатково, сучасні методи оцінки екологічного впливу відображають залежність між енергоспоживанням ІТ-інфраструктури та викидами CO₂, що представлено у таблиці 1.1 [6,7].

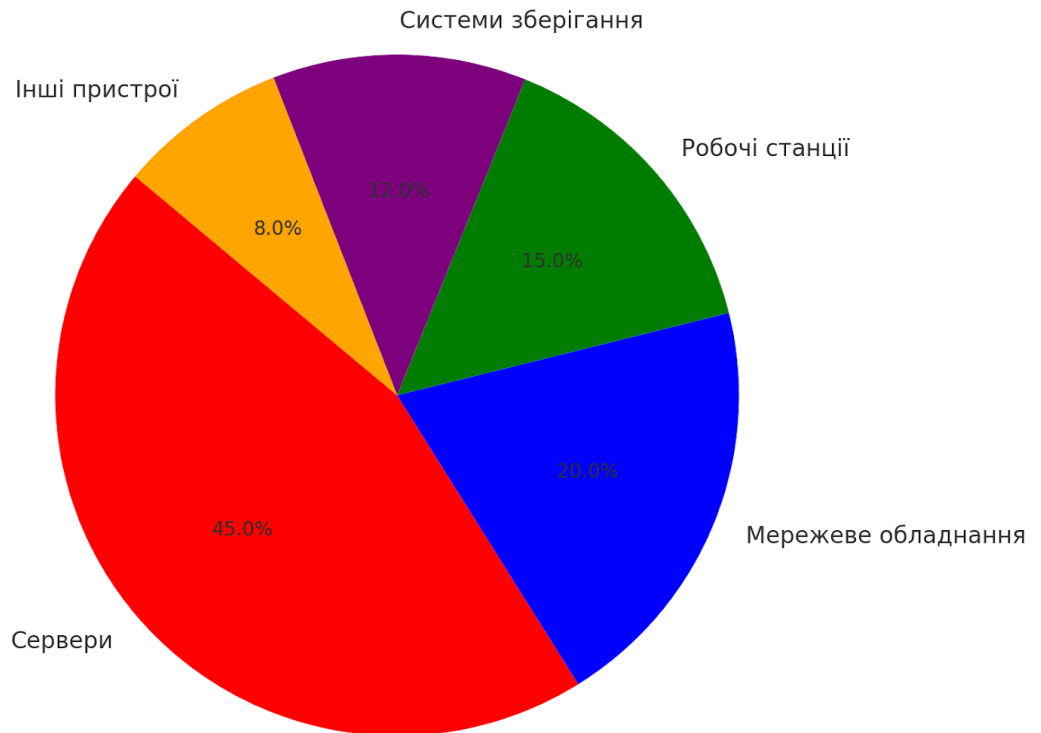


Рисунок 1.1 - Діаграма розподілу енергоспоживання за видами пристроїв у типовій мережі.

Таблиця 1.1 - Залежність між енергоспоживанням ІТ-інфраструктури та викидами CO₂

Джерело споживання	Відсоток від загального	Викиди CO ₂ (т/рік)
Сервери	40%	500,000
Комутатори	30%	300,000
Клієнтські пристрої	20%	200,000
Охолодження	10%	100,000

Ось кілька визначень поняття перенавантаження в комп'ютерних мережах, запропонованих різними авторами.

Cisco визначає перевантаження в мережі як ситуацію, коли "обсяг активності в мережі перевищує її здатність обробляти дані". Це може спричинити зниження продуктивності мережі, перебої в її роботі або вихід обладнання з ладу [8].

За визначенням Microsoft, перевантаження виникає тоді, коли "обсяг даних, які необхідно передати, перевищує пропускну здатність мережі". Це часто призводить до зменшення швидкості передачі даних та виникнення збоїв у роботі мережі [9].

Amazon Web Services (AWS) пояснює перевантаження як ситуацію, коли "рівень навантаження на мережу перевищує її пропускну здатність або продуктивність". Як наслідок, може спостерігатися зниження ефективності роботи додатків та збільшення часу відгуку мережі [10].

Комп'ютерні мережі працюють у різних умовах швидкості передачі даних, що є актуальною проблемою як для Ethernet-мереж, так і для мобільних мереж. Велика кількість мобільних агентів у мережі може спричинити її перевантаження. Одним із підходів до розв'язання цієї проблеми є оптимізація маршрутизації з урахуванням змін у топології мережі [11].

Оскільки універсального алгоритму маршрутизації для комп'ютерних мереж не існує, у запропоновано динамічно розподіляти трафік, використовуючи систему агентів, яка визначає суміжних агентів. У роботі розглянуто питання усунення перевантаження мобільних мереж в умовах навмисних і ненавмисних перешкод. Це досягається шляхом реконфігурації діаграм направленості антен базових станцій, що дозволяє підвищити завадозахищеність стільникового зв'язку [12, 13].

Останні дослідження в цій сфері, зокрема, пропонують методіку побудови оптимізаційних моделей мереж, яка передбачає розміщення вузлів мережевої структури за допомогою електронних таблиць. Вузли поділяються на сервери та прив'язаних до них клієнтів. Конфігурація графу такої мережі формується шляхом розв'язання максимінної задачі з використанням нелінійної оптимізації [14].

Крім того, у запропоновано застосування методу слідкуючого вікна для управління потоками даних у локальних мережах, створених мобільними агентами.

Динамічне керування частотою та напругою (DVFS – Dynamic Voltage and Frequency Scaling). Ця стратегія передбачає регулювання тактової частоти та напруги живлення процесорів (як центральних – CPU, так і графічних – GPU) залежно від поточного або прогнозованого навантаження на систему. Механізм дії: зниження частоти та відповідно напруги при невисокому навантаженні дає змогу скоротити енергоспоживання, тоді як у періоди пікового використання систему можна перевести в режим максимальної продуктивності [15, 16].

Застосування у серверних фермах (кластерних системах) DVFS допомагає підтримувати баланс між енергозбереженням і продуктивністю. Сучасні процесори мають вбудовані технології (Intel SpeedStep, AMD Cool'n'Quiet тощо), що дозволяють здійснювати ці переходи автоматично [17]. Переваги: зменшення тепловиділення й, відповідно, витрат на охолодження. Недоліки: зміни частоти можуть впливати на затримки та вимагати часу на синхронізацію тактових генераторів [18].

У сучасних комп'ютерних мережах однією з ключових цілей є забезпечення високої продуктивності при оптимальному рівні енергоспоживання. Зростання обсягів обчислень та кількості обладнання, необхідного для обробки даних, стимулює пошук методів та стратегій зниження витрат енергії без погіршення показників продуктивності. Нижче наведено основні стратегії керування потужністю систем, які можуть застосовуватися залежно від профілів робочих параметрів [19, 20].

Керування живленням на основі політик (Power Policy Management)

Цей підхід полягає у визначенні заздалегідь сформованих політик або профілів енергоспоживання, які обираються динамічно відповідно до умов експлуатації.

Механізм дії - розклад (Scheduler-based): переведення систем у режим низького енергоспоживання в нічні години або при зменшенні інтенсивності

навантаження[21] . Подієве керування (Event-driven): активація енергозберігаючих механізмів при тривалому простої обладнання або навпаки – повернення у високопродуктивний режим при виявленні пікового навантаження. Застосування - використовується в корпоративних середовищах, де можна налаштувати типові сценарії роботи (офісні години, нічні/вихідні, періоди резервного копіювання та ін.)[22]. Переваги - зручність для централізованого адміністрування, чітка відповідність певним бізнес-процесам чи розкладам. Недоліки: потребує правильного налаштування політик, інакше система може передчасно переходити до зниженого рівня продуктивності або, навпаки, не використовувати весь потенціал енергозбереження [23].

Керування живленням на рівні мережевої інфраструктури (Network Power Management). Окрім обчислювальних ресурсів, значну частку енергії споживає мережеве обладнання: комутатори, маршрутизатори, мережеві інтерфейсні карти (NIC) тощо [24].

Стандарти та технології [25, 26]:

- IEEE 802.3az (Energy Efficient Ethernet) – дозволяє автоматично знижувати споживання енергії портами Ethernet, коли пропускна здатність не використовується повністю.
- Wake-on-LAN – дає змогу віддалено вмикати/вимикати пристрої за потреби.

Методи скорочення споживання: переведення невикористовуваних портів у низькочастотний або сплячий режим [27]. Оптимізація маршрутизації для мінімізації використання надмірного обладнання (портів, лінків). Переваги: зниження загальних витрат енергії інфраструктури та витрат на охолодження. Недоліки: може виникати необхідність швидкого повернення в активний режим, що інколи призводить до збільшення затримок у мережі [28, 29].

Розподілене управління навантаженням (Load Balancing for Power Efficiency). У масштабних центрах обробки даних оптимальний розподіл навантаження між серверами й вузлами дозволяє знижувати кількість «гарячих» точок та уникати ситуацій, коли частина обладнання працює на межі

можливостей, тоді як інша – простоює [30]. Механізм дії - постійне відстеження навантаження на кожному сервері або віртуальній машині [31]. Розподіл ресурсів - передача завдань з перевантажених вузлів на менш завантажені або тимчасово невикористовувані. Відключення (Consolidation) - за можливості – об'єднання завдань на мінімальній кількості вузлів і переведення інших у режим зниженого енергоспоживання (наприклад, “sleep” чи “hibernation”) [32]. Використання віртуалізації (VMware, Hyper-V, KVM) та контейнеризації (Docker, Kubernetes) полегшує перерозподіл обчислювальних ресурсів у реальному часі [33]. Переваги: значна економія енергії при масштабних операціях, можливість швидко адаптуватися до зміни навантаження. Недоліки: потребує складних механізмів планування і моніторингу, можливі короточасні затримки на перенесення завдань [34].

Енергетично ефективні алгоритми планування (Energy-Aware Scheduling). Традиційні системи планування переважно зосереджені на максимальному використанні ресурсів або мінімізації часу обробки задач. Однак енергетично ефективні алгоритми додають ще один вимір – оптимізацію енергоспоживання. Використання евристик (наприклад, Shortest Job First, MinEnergyFirst), що враховують потужність кожного вузла [35]. Застосування методів машинного навчання, щоб передбачити майбутнє навантаження й заздалегідь адаптувати конфігурації. На рівні операційної системи або в системах оркестрації (Kubernetes, Mesos), що можуть планувати ресурси відповідно до профілів енергоспоживання кожного вузла. Переваги: можливість більш тонкого регулювання енергоспоживання, врахування пріоритетності завдань та вимог до часу відгуку. Недоліки: ускладненість реалізації, потреба в актуальних моделях навантаження, висока обчислювальна складність деяких алгоритмів [36, 37].

Гібридні методи поєднують у собі декілька підходів для досягнення максимальної ефективності. Наприклад, система може одночасно застосовувати DVFS, політику відключення неактивних ресурсів та алгоритми динамічної оркестрації навантаження з урахуванням енергоспоживання [38]. Використання

штучного інтелекту (AI) та методів машинного навчання для комплексного прогнозування навантажень і автоматичного застосування кращої стратегії [39].

1.2 Огляд сучасних методів оптимізації енергоспоживання

Сучасні методи оптимізації енергоспоживання включають [40, 41] динамічне регулювання частоти та напруги (DVFS) використовується для зниження споживання енергії за рахунок динамічної адаптації частоти процесорів до поточного навантаження, ця технологія дозволяє економити до 20-30% енергії та віртуалізацію ресурсів, що забезпечує консолідацію робочих процесів на мінімальній кількості фізичних серверів [42].

Алгоритми маршрутизації: Використання енергозберігаючих протоколів маршрутизації, таких як Green Routing Protocol (GRP), дозволяє зменшити енергоспоживання на 15-25% [43].

Інтеграція відновлюваних джерел енергії: Впровадження сонячних батарей і вітрогенераторів у мережеві системи. Це дозволяє компенсувати до 50% загального енергоспоживання [44] у таблиці 1.2 порівняно ефективність зазначених методів [45, 46].

Таблиця 1.2 - Ефективність методів оптимізації енергоспоживання

Метод	Економія енергії	Основні переваги	Недоліки
DVFS	20-30%	Простота реалізації	Необхідність підтримки на апаратному рівні
Віртуалізація	40-50%	Висока ефективність	Високі початкові витрати
Green Routing Protocol	15-25%	Зниження енергоспоживання	Залежність від типу трафіку

Відновлювані джерела	До 50%	Екологічна стійкість	Залежність від погодних умов
----------------------	--------	----------------------	------------------------------

Перевантаження в комп'ютерних мережах може спричинити зниження продуктивності, збої в роботі, затримки у виконанні завдань та інші негативні наслідки. Для запобігання таким проблемам у комп'ютерних мережах впроваджуються спеціальні механізми керування перевантаженням [47].

Керування перевантаженням механізмів у комп'ютерних мережах є важливим аспектом забезпечення їхньої стабільної та ефективної роботи. У сучасних мережах часто виникають ситуації, коли обсяг трафіку або навантаження на сервери перевищує заплановані показники, що може призвести до затримок у передачі даних, збоїв у роботі чи навіть до втрати інформації. Для вирішення цих проблем існує низка механізмів і технологій [48, 49].

Одним із найпоширеніших методів є динамічне балансування навантаження. Цей підхід передбачає розподіл вхідного трафіку між кількома серверами або вузлами мережі. Завдяки цьому один сервер не перевантажується, а інші залишаються недозавантаженими [50]. Наприклад, у таких великих платформах, як Amazon або Netflix, використовуються балансувальники навантаження, що перенаправляють запити користувачів до найменш завантажених серверів. Це забезпечує швидку відповідь системи та рівномірний розподіл ресурсів. Алгоритми, які застосовуються для балансування, можуть бути різними: від простого послідовного розподілу запитів до аналізу часу відповіді кожного сервера і вибору найоптимальнішого [51].

Іншим ефективним методом є використання віртуалізації ресурсів. Ця технологія дозволяє створювати на фізичних серверах кілька віртуальних машин, які здатні динамічно змінювати свої ресурси залежно від навантаження [52]. Наприклад, платформи, такі як VMware vSphere, впроваджують механізм Distributed Resource Scheduler (DRS), який автоматично переносить віртуальні машини між серверами, щоб уникнути перевантаження одного з них. У такий

спосіб досягається оптимальне використання фізичних ресурсів і підвищується загальна стійкість системи [53].

Ще одним важливим підходом є резервування ресурсів. У випадках, коли один із серверів виходить з ладу або перевантажується, система автоматично перемикає трафік на резервний сервер [54]. Цей механізм активно використовується у банківських системах, таких як Visa чи MasterCard, де забезпечення безперебійної роботи є критично важливим. Резервний сервер зазвичай перебуває в режимі очікування і активується тільки у випадку необхідності [55].

Для управління трафіком у вузлах мережі також застосовуються алгоритми контролю черг. Наприклад, у мережевих пристроях Cisco використовується технологія Random Early Detection (RED), яка відстежує завантаженість черг і відкидає некритичні пакети у разі ризику переповнення. Це дозволяє зберегти якість обслуговування (QoS) навіть у пікові періоди [56, 57].

Адаптивна маршрутизація є ще одним ефективним методом управління перенавантаженням. Вона базується на використанні динамічних протоколів маршрутизації, таких як OSPF (Open Shortest Path First), які автоматично обчислюють оптимальні маршрути для трафіку. У разі перевантаження одного з вузлів протоколи перенаправляють трафік через менш завантажені шляхи, що допомагає уникнути затримок і втрат даних [58, 59].

У сучасних розподілених мережах дедалі частіше застосовуються технології CDN (Content Delivery Network), які дозволяють зменшити навантаження на центральний сервер шляхом використання розподілених серверів для доставки контенту. Наприклад, Cloudflare або Akamai CDN зберігають кешовані копії контенту у різних точках світу, доставляючи його до користувача з найближчого сервера. Це не тільки знижує затримки, але й дозволяє основним серверам обробляти менше запитів [60].

Ще одним ефективним підходом є алгоритми обмеження швидкості (Rate Limiting). Ця технологія використовується для контролю швидкості передачі даних, щоб уникнути перевантаження серверів. Наприклад, сервіси API, такі як

Google Maps, обмежують кількість запитів від одного користувача в одиницю часу. Це допомагає уникнути надмірного навантаження на сервери та підтримувати стабільну роботу системи [61].

Нарешті, важливу роль у керуванні перенавантаженням відіграють енергозберігаючі механізми. У сучасних дата-центрах, наприклад у Google, впроваджуються технології, які переводять сервери в режим низького енергоспоживання під час простою. Це не лише знижує витрати на електроенергію, але й зменшує навантаження на систему загалом [62].

Таким чином, керування перенавантаженням у комп'ютерних мережах включає широкий спектр технологій і підходів, які забезпечують стабільність, ефективність та економічність роботи мереж. Кожен із розглянутих методів має свої переваги й обмеження, але їхнє комбіноване використання дозволяє досягти оптимальних результатів навіть у високонавантажених системах [63].

У сучасних умовах, коли енергоефективність та оптимізація виробничих процесів стають ключовими аспектами розвитку технологій, використання профілів робочих параметрів для керування потужністю набуває все більшого значення. Профілі робочих параметрів є спеціально розробленими наборами налаштувань або режимів роботи, які дозволяють системі адаптуватися до різних умов експлуатації. Вони забезпечують оптимальне співвідношення між продуктивністю, енергоспоживанням та надійністю обладнання, що робить їх незамінним інструментом у сучасних системах керування [64, 65].

Профілі робочих параметрів представляють собою комплексні набори налаштувань, які визначають поведінку системи в різних умовах. Вони можуть включати такі параметри, як рівні потужності (мінімальний, номінальний, максимальний), частота роботи обладнання, температурні режими, час активізації певних функцій або процесів [66]. Ці параметри формуються на основі детального аналізу технологічних вимог, енергетичних характеристик обладнання та умов експлуатації. Наприклад, у промислових системах профілі можуть враховувати рівень навантаження, час роботи обладнання та зовнішні фактори, такі як температура навколишнього середовища [67].

Використання профілів робочих параметрів у системах керування потужністю дозволяє досягти значного покращення ефективності роботи обладнання. Однією з ключових переваг є можливість автоматизації процесів керування [68]. Система може автоматично перемикатися між різними профілями в залежності від поточних умов, таких як рівень навантаження, час доби або температура навколишнього середовища. Це дозволяє забезпечити оптимальний режим роботи обладнання без необхідності постійного втручання оператора [69].

Ще однією важливою перевагою використання профілів є підвищення надійності роботи обладнання. Правильний вибір профілів дозволяє уникнути перевантажень, перегріву та передчасного зношування обладнання, що значно збільшує термін його служби та знижує витрати на обслуговування. У енергетичних системах профілі робочих параметрів допомагають балансувати навантаження, оптимізувати роботу електростанцій та інтегрувати відновлювані джерела енергії. Наприклад, у сонячних електростанціях профілі можуть враховувати рівень сонячної активності та автоматично регулювати потужність генерації [70].

Використання профілів робочих параметрів у керуванні потужністю має ряд значних переваг. По-перше, це гнучкість системи, яка дозволяє швидко адаптуватися до змінних умов експлуатації. По-друге, це економія ресурсів, оскільки правильний вибір профілів дозволяє знизити витрати на енергію та обслуговування обладнання [71]. По-третє, це підвищення продуктивності, оскільки оптимізація роботи системи за рахунок вибору найефективнішого режиму дозволяє досягти максимальної ефективності. Нарешті, це надійність, оскільки правильний вибір профілів допомагає уникнути перевантажень та передчасного зношування обладнання [72, 73].

Для реалізації профілів робочих параметрів використовуються сучасні засоби автоматизації та керування. Одним із ключових елементів є програмовані логічні контролери (PLC), які дозволяють автоматично вибирати та перемикати профілі в залежності від поточних умов [74]. Також широко використовуються

SCADA-системи, які забезпечують моніторинг та управління процесами в реальному часі.

Для збору даних про поточні умови роботи та навантаження використовуються різні датчики та системи моніторингу. На основі цих даних спеціальні алгоритми оптимізації вибирають найефективніший профіль роботи. Наприклад, у промислових системах це може бути алгоритм, який аналізує рівень навантаження та температуру навколишнього середовища, і на основі цих даних вибирає оптимальний режим роботи обладнання [75, 76].

Використання профілів робочих параметрів у керуванні потужністю є сучасним підходом, який забезпечує ефективне використання енергетичних ресурсів, підвищення продуктивності та надійності роботи систем [77]. Цей метод знаходить широке застосування в промисловості, енергетиці та побутових системах, що робить його важливим елементом сучасних технологій керування. Завдяки своїй гнучкості, економічності та надійності, профілі робочих параметрів стають незамінним інструментом для оптимізації роботи систем керування потужністю [78].

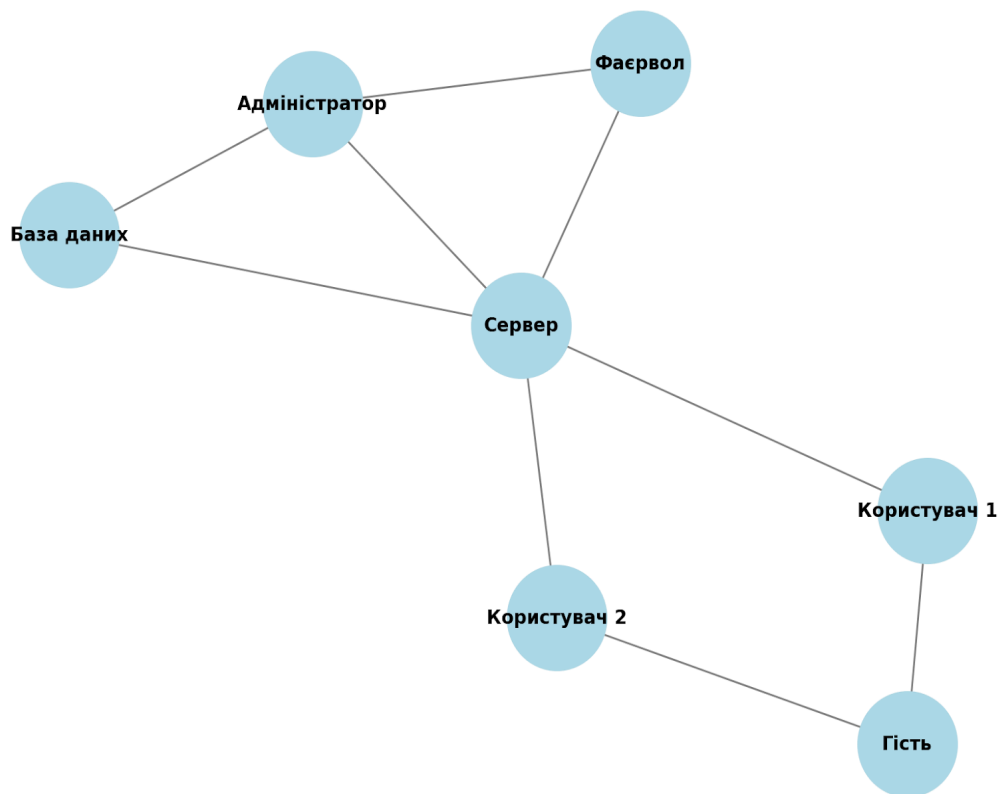


Рисунок 1.2 - Структура профілів у комп'ютерній мережі

Профілі робочих параметрів (рисунок 1.2) забезпечують адаптацію режимів роботи обладнання до змін умов експлуатації. Часові профілі: визначають режими роботи залежно від часу доби (піковий/непіковий час). Профілі навантаження: автоматично регулюють потужність залежно від інтенсивності трафіку. Специфічні профілі: орієнтовані на виконання окремих завдань, наприклад обробка відео [79].

Адміністратор має доступ до сервера, бази даних та фаєрвола, керуючи всіма основними ресурсами. Сервер взаємодіє з базою даних, а також обслуговує користувачів. Користувачі (Користувач 1 та Користувач 2) отримують доступ до сервера та можуть взаємодіяти з гостьовим доступом. Фаєрвол забезпечує контроль трафіку між сервером та мережею [80].

1.3 Постановка задачі

Метою роботи є розробка методу керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів, що дозволить підвищити ефективність використання енергетичних ресурсів та забезпечити стабільність функціонування системи при зміні навантаження.

Поставлена мета досягається виконанням таких основних завдань:

- проаналізувати існуючі методи керування потужністю в комп'ютерних мережах, орієнтовані на використання профілів робочих параметрів;
- розробити метод керування потужністю, що базується на аналізі та кластеризації вузлів мережі відповідно до профілів їх робочих параметрів та поточного завантаження;
- визначити критерії оцінювання ефективності запропонованого методу керування потужністю;
- реалізувати програмний прототип запропонованого методу та провести експериментальні дослідження для оцінки його ефективності.

1.4 Висновки

У розділі розглянуто значення оптимізації енергоспоживання в комп'ютерних мережах через методи, такі як динамічне регулювання частоти (DVFS), віртуалізація ресурсів та адаптивні алгоритми. Інтеграція різних стратегій керування потужністю дозволяє зменшити експлуатаційні витрати, екологічний вплив і підвищити стабільність та продуктивність мережевих систем. Особливо ефективним є використання методів машинного навчання для управління енергоспоживанням у реальному часі.

2 ОСНОВИ ДО КЕРУВАННЯ ПОТУЖНІСТЮ В КОМП'ЮТЕРНИХ МЕРЕЖАХ ЗА РІЗНИМИ РОБОЧИМИ ПАРАМЕТРАМИ

2.1 Методи аналізу енергоспоживання вбудованих систем на ранніх етапах проектування

Під час розробки нових систем Інтернету речей (IoT) критично важливо з самого початку проектування враховувати енергоспоживання, особливо при створенні кінцевих вузлів, які працюють в умовах обмежених енергетичних ресурсів. Це актуально для пристроїв, що живляться від батарей або використовують технології збору енергії з навколишнього середовища. Тому оцінювання енергоспоживання має здійснюватися на ранніх етапах проектного циклу, оскільки це дозволяє не лише сформувати енергоефективну архітектуру, але й зменшити витрати часу та фінансових ресурсів.

Загальне енергоспоживання електронної системи складається з двох основних компонентів - статичної та динамічної складових. Динамічна складова включає енергію, яка витрачається на перемикання станів транзисторів (так зване комутаційне споживання) та потужність короткого замикання, що виникає під час переходу між логічними станами, коли водночас активні і високий, і низький рівень.

Комутаційна потужність (PT) обчислюється за формулою:

$$PT = C_{pd} \times V_{cc}^2 \times f_I \times N_{SW} \quad (2.1)$$

де C_{pd} — динамічна ємність споживання потужності,

V_{cc}^2 — напруга живлення,

f_I — частота сигналу,

N_{SW} — кількість перемикань (бітів).

Потужність короткого замикання (P_{SC}) визначається за інтегралом:

$$P_{SC} = \int_{t1}^{t3} V_{dd} \times I_{SC}(t) dt = \frac{1}{2}(t3 - t1) \times I_{SC}^{max} \times V_{dd}, \quad (2.2)$$

де V_{dd} - напруга живлення,

$I_{SC}(t)$ - струм короткого замикання в момент часу t ,

I_{SC}^{max} - максимальний струм короткого замикання,

$t1$ і $t3$ - початок та завершення фази короткого замикання відповідно.

Енергоспоживання системи загалом визначається як інтеграл потужності за часом. Для зменшення цього показника розробники застосовують стратегії керування енергоспоживанням, що дають змогу адаптивно перемикатися між енергетичними режимами залежно від поточних вимог до продуктивності. На ранніх етапах проєктування здійснюється оцінка енергоспоживання на системному рівні, однак більшість традиційних підходів не враховують ефектів керування живленням, що призводить до неточностей. З іншого боку, оцінювання на нижчих рівнях абстракції (наприклад, на рівні RTL або транзисторів) є точнішим, але неефективним для складних систем через високі витрати на зміну конструкції.

У зв'язку з цим у статті запропоновано новий метод оцінки динамічного енергоспоживання на системному рівні з урахуванням керування живленням. Рішення реалізовано у вигляді інструменту "Power Estimation at System Level" (PESL), який дозволяє обчислити відносне енергоспоживання компонентів системи відповідно до заданої стратегії енергоменеджменту. Це забезпечує можливість порівняння декількох варіантів архітектур і вибору найоптимальнішої з погляду енергоефективності.

Хоча в сучасних мікроелектронних системах статичне енергоспоживання також відіграє значну роль, у межах цієї роботи основна увага приділяється саме динамічній складовій, тоді як аналіз статичних аспектів викладено в окремому дослідженні.

На сьогодні існує низка методів оцінки динамічного енергоспоживання на системному рівні, які ґрунтуються на моніторингу активності компонентів і використанні моделей потужності. Це зумовлено прямою залежністю динамічної потужності від частоти сигналів і кількості перемикачів у системі.

Одним із традиційних підходів є інструкційне моделювання, при якому розрахунок енергоспоживання здійснюється на основі попередньо створених таблиць потужності для кожної інструкції. Незважаючи на високу точність, цей метод вимагає попередньої реалізації компонентів і не є ефективним для проектування за підходом «згори донизу».

Аналогічно працює транзакційне моделювання (TLM), де замість інструкцій використовуються транзакції або завдання. Такі методи також залежать від попередніх моделей і не придатні для ранніх етапів проектування.

Більш універсальним є підхід, заснований на відстані Хеммінга, який дозволяє оцінити активність сигналів за кількістю побітових змін. Він може застосовуватись на різних рівнях абстракції та забезпечує достатню точність на системному рівні.

Ще один підхід до адаптації методу це - використання макромоделей потужності, які поєднують дані про активність компонентів із характеристиками їх енергоспоживання. Однак ці методи часто не враховують керування живленням, що обмежує їхню точність.

Інструменти, засновані на стандартах UPF (Unified Power Format), дозволяють описувати домени та стани живлення, але мають низький рівень абстракції, потребують ручної роботи та попередньої реалізації компонентів. Вони рідко застосовуються у повністю автоматизованих проектних потоках.

Перспективним є використання бібліотеки керування живленням (PMS), що дозволяє модифікувати моделі SystemC для опису різних режимів живлення. Попри зручність специфікації, для точної оцінки потужності такі методи потребують аналізу на нижчих рівнях, що ускладнює швидке дослідження альтернативних варіантів керування живленням.

2.2 Метод оцінки енергії на системному рівні

На основі аналізу існуючих підходів розроблено власний метод оцінки енергоспоживання на системному рівні з урахуванням впливу керування живленням. У його основі — вдосконалений розрахунок відстані Хеммінга, який адаптовано до поточних енергетичних станів модулів.

Метод працює наступним чином: на початковому етапі здійснюється обробка даних із двох джерел — функціональної моделі, описаної мовою SystemC, та файлу зміни значень сигналів (VCD), отриманого в результаті симуляції. Після цього алгоритм зчитує VCD-файл і обчислює відстань Хеммінга, що визначає кількість змін бітів у сигналі за певний проміжок часу.

Паралельно здійснюється аналіз структури моделі SystemC, у межах якого ідентифікуються модулі, домени живлення, стани та режими споживання енергії. Для кожного модуля алгоритм визначає його активний енергетичний стан, після чого розрахована відстань Хеммінга коригується з урахуванням відповідного коефіцієнта, який залежить від цього стану. Ці значення підсумовуються для визначення загального енергоспоживання системи. Процес триває до завершення зчитування даних із VCD-файлу.

Для реалізації обробки даних у моєму методі використовується попереднє зчитування всієї ієрархії модулів, портів і змінних, що містяться у файлі VCD. Всі знайдені змінні хешуються у вигляді пари «модуль + змінна», що дозволяє ефективно відстежувати їхній попередній стан. У моделі SystemC визначається верхній модуль, який містить специфікацію енергоменеджменту, включаючи домени та режими живлення. Також обчислюється коефіцієнт для стану DIFF_LEVEL як середнє арифметичне між станами NORMAL та DIFF_LEVEL.

Після завершення обробки даних мій алгоритм переходить до етапу оцінки енергоспоживання. Початково ініціалізуються всі змінні як векторні двійкові значення. Для кожного модуля визначається його поточний стан живлення. Відповідно до цього стану алгоритм або враховує зміну значень змінних у підсумковій відстані Хеммінга, або ігнорує їх. Таким чином, на основі активного

стану модуля здійснюється накопичення даних про енергоспоживання всієї системи.

У разі, коли компонент перебуває у стандартному (нормальному) режимі живлення, він отримує напругу від основного джерела, а його тактова частота залишається незмінною. У такому випадку відстань Хеммінга між попереднім та поточним станами змінно обчислюється за базовим алгоритмом.

Наприклад, якщо попередній стан дорівнює 110011, а поточний — 100111, то відстань Хеммінга становить 2, оскільки два біти змінили своє значення. Це значення додається до загальної суми енергоспоживання за такою формулою:

$$Total_sum = Total_sum + distance \quad (2.1)$$

Коли компонент працює у режимі, що передбачає змінені рівні напруги або частоти (порівняно зі станом NORMAL), динамічне енергоспоживання залежить від обох цих параметрів. У цьому випадку обчислена відстань Хеммінга додатково коригується шляхом множення на коефіцієнт, що представляє середнє арифметичне співвідношення між відповідними рівнями напруги й частоти в режимах DIFF_LEVEL та NORMAL:

$$C_{DIFF_LEVEL} = \frac{1}{2} \left(\frac{V_{DIFF_LEVEL}}{V_{NORMAL}} + \frac{f_{DIFF_LEVEL}}{f_{NORMAL}} \right). \quad (2.3)$$

Тоді підсумкове значення енергоспоживання оновлюється за формулою:

$$Total_sum = Total_sum + distance \times C_{DIFF_LEVEL} \quad (2.4)$$

У режимі очікування (HOLD) компонент залишається під напругою, однак його активність призупинена, тобто тактова частота дорівнює нулю. У цьому стані зміна значень змінних ігнорується, оскільки внутрішній стан компонента не

повинен змінюватися. Відповідно, відстань Хеммінга не розраховується, а попередній стан змінної залишається незмінним. Загальне значення енергоспоживання також не змінюється.

У разі переходу компонента в стан вимкнення (OFF), він повністю деактивується. Хоча активність у цьому стані не враховується, важливо правильно обробити перехідні процеси. Для цього при вході у стан OFF виконується розрахунок відстані Хеммінга між попереднім станом змінної та умовним нульовим вектором. Після цього 0-вектор зберігається як новий попередній стан.

При виході зі стану OFF знову виконується обчислення відстані Хеммінга — цього разу між 0-вектором та поточним станом змінної. Обидва значення додаються до загальної суми енергоспоживання:

$$Total_sum += distance_{enter_OFF} + distance_{leave_OFF} \quad (2.5)$$

У зазначеному режимі живлення компонент деактивується, однак його внутрішній стан зберігається. Це свідчить про відсутність активності в межах компонента під час перебування у відповідному стані. Як наслідок, алгоритм, подібно до випадку зі станом HOLD, не виконує обчислення відстані Хеммінга. У результаті, внесок цього режиму в загальне енергоспоживання відсутній, і відповідна величина не змінюється.

Після завершення обчислення сумарної відстані Хеммінга для кожного компонента отримане значення додається як до загального енергоспоживання системи, так і до показників споживання конкретного екземпляра модуля, реалізованого в SystemC.

Огляд запропонованого методу оцінки потужності, а також схема вхідних і вихідних даних для кожного з його етапів наведено на рисунку 2.1.

На зображенні представлена блок-схема, яка демонструє принцип роботи алгоритму для аналізу енергоспоживання в комп'ютерній системі. В якості вхідних даних використовуються два джерела: файл VCD, що містить назви

модулів, індикатори зміни режиму живлення, хешовані зміни та змодельовану активність системи, а також файли SystemC, які надають інформацію про режими та домени живлення, а також коефіцієнти множення для кожного режиму. Ці дані обробляються алгоритмом, який на їх основі виконує розрахунки. Результатом роботи алгоритму є вихідні дані, представлені у вигляді блоку «Показати результати». Схема відображає логіку перетворення вхідних параметрів у результат оцінки енергоспоживання.



Рисунок 2.1 - Схема вхідних та вихідних даних.

2.3 Архітектура розробленого інструменту

Запропонований мною метод було реалізовано у вигляді зручного у використанні програмного засобу, основною метою якого є оцінка ефективності методу та демонстрація його переваг на практиці. Алгоритм роботи інструмента передбачає послідовне виконання таких етапів: після запуску користувач завантажує необхідні вхідні дані — файли опису моделі SystemC та файл VCD. У

налаштуваннях передбачено можливість вибору каталогу збереження результатів, формату вихідного файлу, а також внесення змін до інших параметрів обробки.

Після завантаження вхідних файлів запускається автоматизований процес обробки даних, на основі якого ініціюється розрахунок енергоспоживання. Хід виконання обчислень відображається за допомогою індикатора прогресу у відсотковому значенні, а підсумкові результати автоматично експортуються у формат, заданий у налаштуваннях.

Функціональна структура інструмента включає шість окремих блоків, взаємозв'язки між якими наведено на рисунку 2.2. Така модульна архітектура була обрана свідомо — з метою забезпечення гнучкості системи та спрощення її майбутнього розширення.

Модуль «Обробка даних» (Process data) відповідає за аналіз інформації, отриманої з файлів формату VCD та описів SystemC. На початковому етапі своєї роботи він отримує вхідні дані від модуля «Завантаження даних», здійснює їх обробку, після чого передає проаналізовану інформацію назад для подальшого використання. Перед початком обробки цей модуль також приймає параметри конфігурації від модуля «Налаштування».

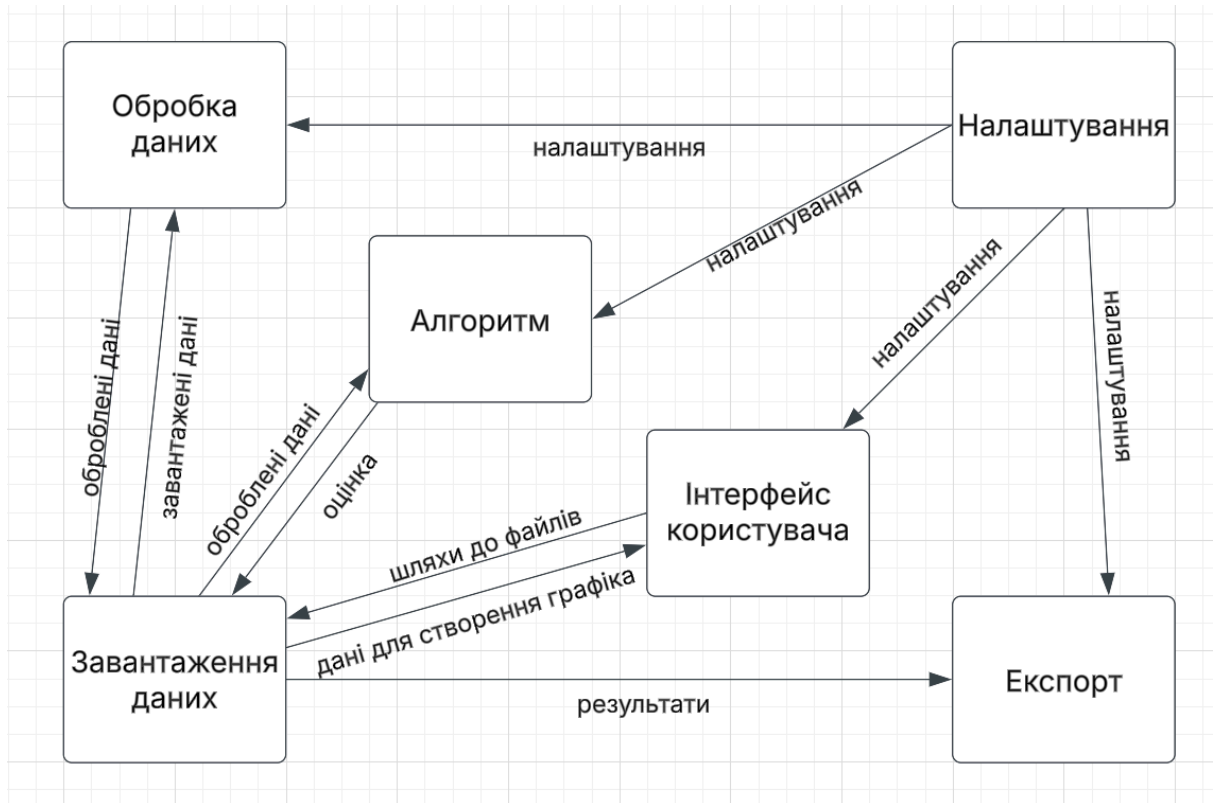


Рисунок 2.2 - Архітектура розробленого інструменту

Модуль «Налаштування» (Config) забезпечує формування та збереження конфігурації інструменту. Він містить типові параметри за замовчуванням, однак за потреби користувач може змінити їх у відповідному вікні інтерфейсу. Оновлені налаштування зберігаються та передаються до всіх модулів автоматично на початку роботи системи.

Модуль «Завантаження даних» (Load data) виконує функцію первинного зчитування даних із зовнішніх джерел, зокрема файлів VCD і SystemC. Крім того, цей модуль відповідає за збереження усіх етапів обробки — зберігає як завантажені, так і проаналізовані дані, передає їх до обчислювального модуля, а також накопичує результати розрахунку енергоспоживання для подальшої передачі в модуль «Експорт».

Модуль «Експорт» (Export) реалізує функцію виведення результатів у різних форматах - XML, PDF, TXT, PNG - та відображення у графічному інтерфейсі. Він працює на основі актуальних параметрів конфігурації або змінених у процесі

використання інструменту налаштувань. За замовчуванням дані експортуються безпосередньо до інтерфейсу користувача.

Модуль «Алгоритм» (Algorithm) відповідає за основні обчислення. Він приймає оброблені дані, виконує розрахунок енергоспоживання відповідно до описаної методики, після чого передає результати назад у модуль «Завантаження даних».

Модуль «Інтерфейс користувача» (GUI) є засобом взаємодії користувача з програмним забезпеченням. Через нього здійснюється завантаження файлів SystemC та VCD, контроль за процесом обробки, перегляд результатів і зміна налаштувань. Передбачено також індикатор виконання, що візуально демонструє прогрес обчислень.

Обмін даними між модулями відбувається за допомогою спеціалізованих інтерфейсів. Дані передаються від одного модуля до іншого, після чого очищаються з пам'яті попереднього, що забезпечує ефективне використання ресурсів системи та зменшує ймовірність дублювання інформації.

Алгоритм (рисунки 2.3) розпочинається з ініціалізації процесу, підготовки середовища та запуску системи. На початковому етапі система завантажує конфігураційні параметри з модуля Config. Це можуть бути як стандартні налаштування за замовчуванням, так і змінні користувачем параметри, збережені з попереднього сеансу. Після завантаження конфігурації ініціалізується графічний інтерфейс користувача. У GUI користувач може керувати процесом, обирати вхідні файли та переглядати результати.

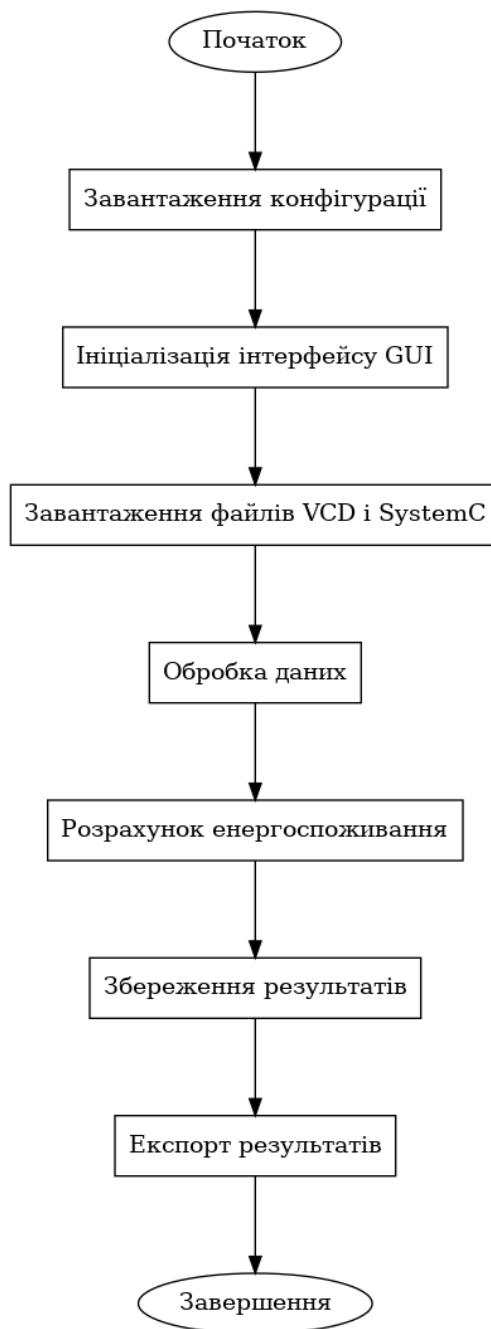


Рисунок 2.3 - Алгоритм розробленого інструменту

Користувач за допомогою інтерфейсу завантажує необхідні вхідні дані, а саме: файл зміни значень сигналів (VCD) та файл опису системи (SystemC). Модуль Process data аналізує завантажену інформацію, визначає ієрархію модулів, змінні, стани живлення та енергетичні домени. Оброблені дані готуються для розрахунку. Модуль Algorithm отримує попередньо оброблені дані і виконує розрахунок динамічного енергоспоживання. Основою є метод відстані Хеммінга з

урахуванням станів живлення (NORMAL, DIFF_LEVEL, HOLD, OFF тощо).

Результати розрахунків повертаються в модуль Load data, де тимчасово зберігаються у внутрішній структурі. Модуль Export отримує дані та експортує результати в один або кілька форматів: XML, PDF, TXT, PNG. Також передбачено автоматичне виведення у графічний інтерфейс.

Після завершення обробки та експорту результатів алгоритм переходить у кінцевий стан, очікуючи нових дій користувача або завершення роботи інструменту.

2.4 Висновки

У цьому розділі було розглянуто важливість врахування енергоспоживання на ранніх етапах проектування систем Інтернету речей (IoT), зокрема при створенні кінцевих вузлів, що працюють в умовах обмежених енергетичних ресурсів. Як показано в роботі, енергоспоживання є критичним аспектом, оскільки пристрої, що живляться від батарей або використовують технології збору енергії, потребують ефективних рішень для зменшення споживаної потужності без шкоди для їх продуктивності.

Зазначено, що динамічне енергоспоживання є важливою складовою, яка включає комутаційну потужність та потужність короткого замикання, що виникають під час переходів між логічними станами. Водночас було вказано на необхідність оцінки енергоспоживання на системному рівні, адже традиційні методи часто не враховують ефекти керування живленням, що може призвести до неточностей у розрахунках.

Запропоновано новий метод оцінки динамічного енергоспоживання з урахуванням керування живленням, що базується на вдосконаленому підході до розрахунку відстані Хеммінга. Цей метод дозволяє точніше оцінити енергоспоживання на різних етапах життєвого циклу системи, забезпечуючи можливість порівняння архітектур та вибору оптимальних варіантів енергоефективності.

Завдяки використанню інструменту "Power Estimation at System Level" (PESL) забезпечується зручна оцінка енергоспоживання компонентів системи за заданою стратегією енергоменеджменту, що є важливим для вибору ефективних рішень на етапі проектування. Підхід дозволяє значно скоротити час на розробку та знизити витрати, що пов'язані з експлуатацією та енергоефективністю.

Загалом, використання цього методу дозволяє більш точно визначати вплив керування живленням на енергоспоживання та забезпечує більш ефективну та економічну розробку систем IoT. Це підкреслює важливість інтеграції енергоефективних стратегій на ранніх етапах проектування, що дозволить забезпечити стабільну роботу кінцевих пристроїв з обмеженими ресурсами енергії.

3 МЕТОД КЕРУВАННЯ ПОТУЖНІСТЮ З УРАХУВАННЯМ ЗМІННИХ ПАРАМЕТРІВ МЕРЕЖІ

Було проаналізовано та реалізовано підхід до управління енергетичним станом окремого компонента або всієї системи в цілому. Компоненти системи розглядаються як «чорні ящики», поведінка яких визначається моделлю керування живленням (Power State Model, PSM). Основну увагу зосереджено на створенні ефективних політик енергозбереження.

Для спрощення задачі фокус був на управлінні одним компонентом системи, що дозволяє розглядати ситуацію в межах динамічного керування живленням (Dynamic Power Management, DPM). У разі ідеалізованих умов — миттєвих переходів між енергетичними станами без додаткових витрат — оптимальною стратегією є жадібна політика: щойно система простоє, її необхідно одразу переводити в найглибший доступний режим енергозбереження з негайним поверненням у активний стан при надходженні запиту.

Проте в реальних умовах перехід між енергетичними станами супроводжується витратами часу та енергії. Наприклад, вимкнення живлення та його повторне вмикання вимагають часу на стабілізацію джерела живлення, запуск тактового генератора, повторну ініціалізацію компонентів та відновлення контексту. З огляду на ці чинники, задача визначення оптимальної стратегії DPM набуває характеру складної оптимізаційної проблеми. Вона передбачає обґрунтоване прийняття рішень про доцільність і час переходу в один із доступних енергозберігаючих режимів з урахуванням впливу на енергоспоживання та продуктивність системи.

3.1 Оптимізація політик динамічного керування живленням (DPM)

У процесі розробки посилену увагу приділено на особливості додатків подіє-орієнтованого типу, таких як сервери відображення, елементи графічного інтерфейсу користувача та комунікаційні інтерфейси. Їх характерною ознакою є

періодична обчислювальна активність, що виникає виключно у відповідь на зовнішні події. У проміжках між такими подіями система зазвичай перебуває у стані очікування або повного простою.

Ця особливість відкриває можливість ефективного зниження середнього енергоспоживання за рахунок тимчасового вимкнення компонентів системи в періоди бездіяльності.

У рамках реалізованого підходу розглянуто методи зупинки системного тактового генератора, а також, у певних випадках, повне відключення джерела живлення відображено на рисунку 3.1. Такий підхід дозволяє значно зменшити енергоспоживання без шкоди для функціональності системи.

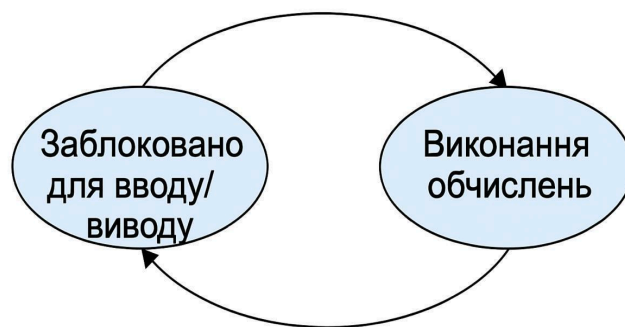


Рисунок 3.1 – Подія-орієнтований застосунок чергується між заблокованим і активним станами.

У розробленому підході подіє-орієнтовані застосунки розглядаються як такі, що циклічно переходять між двома станами: заблокованим — коли додаток очікує зовнішніх подій і фактично перебуває у стані простою, та активним — коли виконує обчислення у відповідь на ці події.

Середні часові характеристики цих станів позначаються як $T_{blocked}$ (перебування у стані очікування) та $T_{running}$ (період активної роботи).

В межах реалізованої моделі показано, що за умови вимкнення системи щоразу, коли вона переходить у стан блокування, можна досягти покращення енергоефективності до теоретичного коефіцієнта:

$$1 + \frac{T_{blocked}}{T_{running}}, \quad (3.1)$$

де $T_{blocked}$ - перебування у стані очікування)

$T_{running}$ (період активної роботи).

Це означає, що що більше система простоє у заблокованому стані порівняно з часом активної роботи, то більший потенціал енергозбереження.

Однак для реалізації такої стратегії необхідно вирішити два основних завдання:

- яким чином здійснювати вимкнення системи - з урахуванням можливих режимів (зупинка тактового генератора, відключення живлення тощо) та відповідних витрат енергії й часу на повернення до активного стану;

- коли саме ініціювати перехід до енергозберігаючого режиму - тобто як визначити оптимальний момент для зниження енергоспоживання без критичного впливу на продуктивність системи.

У межах реалізованого підходу перше з ключових питань - як саме вимикати систему — вирішується шляхом розробки механізмів, що дозволяють зупиняти та повторно запускати тактовий генератор або повністю вимикати джерело живлення з можливістю подальшого його автоматичного підключення. Друге питання - коли саме ініціювати вимкнення - розв'язується через формулювання відповідних стратегій, наприклад: «вимкнути систему, якщо не виявлено активності користувача протягом п'яти хвилин».

Ці два аспекти тісно пов'язані між собою, оскільки ефективність стратегії залежить від витрат енергії та часу, необхідних для входу в енергозберігаючий стан і повернення до активного режиму. Проте в рамках мого дослідження акцент зроблено саме на визначенні моменту вимкнення, що стало можливим завдяки використанню методів прогнозування, які базуються на історії попередніх циклів обчислень.

Прості стратегії, такі як автоматичне вимкнення дисплея чи переходу системи в сплячий режим після кількох секунд/хвилин неактивності, давно використовуються в сучасних ноутбуках і десктопах. Проте вивчено можливість застосування більш гнучких та агресивних стратегій для подіє-орієнтованих систем. Завдяки сучасним апаратним засобам керування живленням (Power Management Controllers, PMC) стало можливим тимчасове вимкнення навіть окремих підсистем на дуже короткий час — у проміжках між подіями.

У запропонованій реалізації механізм керування вимкненням ґрунтується на прогнозуванні тривалості простою центрального процесора. На основі аналізу попередньої історії його активності визначається ймовірна тривалість наступного періоду бездіяльності. Якщо очікуваний час простою перевищує певний поріг, що компенсує витрати на вхід/вихід з енергозберігаючого режиму, приймається рішення про вимкнення процесора. Такий підхід дозволяє досягти ефективного балансу між економією енергії та збереженням продуктивності системи.

У основі прогнозного підходу до енергозбереження лежить проста, але потужна ідея: «Використовувати історичні дані для передбачення, чи буде наступний період простою ($T_{blocked}$) достатньо довгим, щоб виправдати вимкнення системи.

Проте на практиці точне передбачення тривалості простою є складним завданням, схильним до помилок через нестабільність і випадковість надходження подій. Саме тому у реальних системах широко застосовуються евристичні методи оцінки, які базуються на нещодавній історії активності системи.

У межах мого дослідження були проаналізовані підходи, що дозволяють на основі поточної динаміки навантаження оцінити, чи буде наступний період бездіяльності достатньо довгим для компенсації витрат на входження та вихід з енергозберігаючого режиму.

Результати моделювання демонструють, що навіть прості прогнозні стратегії дозволяють досягти суттєвого зменшення енергоспоживання порівняно з

«жадібними» підходами (де система вимикається негайно після настання простою), при цьому практично не погіршуючи загальну продуктивність.

Для ілюстрації ефективності цього підходу розглянуто спрощену модель подіє-орієнтованого застосунок, що працює на базі процесора SA-1100. У даній моделі процесор самостійно перемикається між станами RUN (активна фаза) та STDBY (очікування) залежно від навантаження. Водночас перехід у стан SLEEP (глибокий сон) ініціюється зовнішніми подіями або відповідним програмним забезпеченням.

Загальну поведінку такої системи доцільно змодельовати як частково автономну систему керування енергоспоживанням (Power State Machine, PSM), яка має лише два макростани (рисунок 3.2):

- ON - об'єднує як RUN, так і STDBY, між якими процесор переходить локально, без участі зовнішніх контролерів;
- OFF - відповідає стану SLEEP, активація якого пов'язана з більшими витратами часу та енергії.

Очікуване енергоспоживання в макростані ON розраховується як середнє значення з урахуванням ймовірностей переходів між RUN та STDBY і тривалості перебування в кожному з них. Переходи між ON і OFF моделюють перемикання між активним режимом і режимом глибокого сну, при цьому накладні витрати на ці перемикання (енергія та затримка) враховуються окремо.

Таким чином, прогнозний підхід у поєднанні з моделлю PSM дозволяє здійснювати інтелектуальне керування живленням у подіє-орієнтованих системах, адаптуючи політику вимкнення до реального навантаження та історії активності системи.

Процесор починає свою роботу в стані ON і здійснює переходи від ON до OFF, а потім знову до ON. Нехай $T_{ON}[i]$ та $T_{OFF}[i]$ позначають час, який застосунок проводить у i -й візит до станів ON і OFF відповідно. Далі визначимо T_{ON} як середнє значення $T_{ON}[i]$ по всіх i , а T_{OFF} як середнє значення $T_{OFF}[i]$ по всіх i . Нехай $T_{(ON-OFF)}$ та $E_{(ON-OFF)}$ позначають час та

енергетичні витрати, пов'язані з переходом у стан OFF. Аналогічно визначаються $T_{(OFF-ON)}$ та $E_{(OFF-ON)}$.

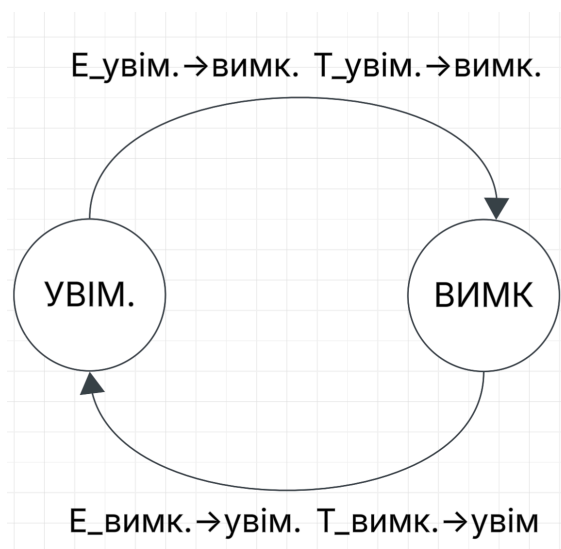


Рисунок 3.2 – PSM (система керування енергоспоживанням) компонента з двома станами.

У підходах до прогнозованого вимкнення система керування енергоспоживанням (PM) прогнозує тривалість часу, протягом якого система буде бездіяльною ($T_{OFF}[i]$), на основі інформації про поточний активний період $[T_{ON}[i]$ та попередні активні та бездіяльні періоди $T_{ON}[j]$ та $T_{OFF}[j]$ для $j=i-1, i-2, \dots, 1$. Політика полягає в тому, щоб перевести процесор з ON у OFF стан, якщо $T_{OFF}[i] \geq T_{BE}$, де T_{BE} - це тривалість часу, необхідного для досягнення точки беззбитковості (break-even time). Процесор знову включається (переходить з OFF в ON стан), коли надходить новий запит на обробку даних.

Були запропоновані кілька інших адаптивних прогнозних методик, що дозволяють ефективніше працювати з нестабільними (нестационарними) навантаженнями.

У межах реалізованого підходу до прогнозування тривалості періодів простою використано ідею динамічного вибору найефективнішого прогнозу на основі результатів попередніх спостережень. Зокрема, було запроваджено набір можливих значень прогнозу тривалості простою, кожне з яких супроводжується

індикатором ефективності — умовною оцінкою того, наскільки вдало цей прогноз працював би в попередніх ситуаціях.

Під час роботи системи ці індикатори постійно оновлюються на основі фактичної тривалості простоїв, що спостерігались раніше. Таким чином, модель адаптується до змін у поведінці системи. Для оцінки наступного періоду бездіяльності вибирається той прогноз, що має найвищий індикатор ефективності, тобто виявився найнадійнішим на основі попередніх циклів.

Такий підхід дозволяє досягти балансу між швидкістю адаптації до нових умов та стійкістю до випадкових коливань, забезпечуючи високу якість прогнозування при низьких обчислювальних витратах.

Альтернативний підхід до прогнозування тривалості простоїв, реалізований у межах мого дослідження, базується на використанні зваженого набору кандидатів-прогнозів, аналогічно до методу, запропонованого Хелмболдом.

У цій моделі кожному можливому значенню тайм-ауту (тобто прогнозу тривалості простою) призначається вага, що відображає його відносну ефективність порівняно з теоретичною, ідеальною офлайн-стратегією.

На основі накопиченої статистики система оцінює, наскільки близьким був кожен кандидат до оптимального варіанту в попередніх сценаріях. Ці оцінки постійно оновлюються, дозволяючи моделі гнучко адаптуватися до змін у характері навантаження.

Фінальне прогнозне значення для наступного періоду простою обчислюється як зважене середнє всіх кандидатів, що дозволяє врахувати накопичений досвід, уникнути жорстких рішень та зменшити вплив окремих хибних прогнозів.

Такий підхід забезпечує стабільнішу роботу системи керування живленням за умов нерегулярної активності застосунку й дозволяє досягти високого рівня енергоефективності без суттєвого зниження продуктивності.

Ще одним варіантом реалізації прогнозного підходу, який досліджено у межах роботи, є адаптивна стратегія з єдиним значенням прогнозу, що постійно коригується в залежності від результатів своєї роботи. Цей підхід натхненний

методом, запропонованим Дугласом.

У цій моделі система не використовує множину кандидатів-прогнозів, натомість оперує одним адаптивним тайм-аутом, який змінюється в процесі роботи:

- якщо поточне значення призводить до надмірної кількості вимкнень, що не виправдовуються реальними простоями (тобто система часто вимикається «зайве»), тайм-аут збільшується;
- якщо ж система надто рідко переходить у режим енергозбереження, втрачаючи можливості для економії енергії, тайм-аут зменшується.

Таким чином, система автоматично підлаштовується до характеру навантаження, змінюючи свою чутливість до періодів бездіяльності залежно від результатів попередніх рішень. Цей підхід є простим у реалізації та ефективним у сценаріях із відносно стабільною поведінкою застосунку, де постійна адаптація одного параметра дозволяє швидко знаходити компроміс між частотою вимкнень і продуктивністю.

Точність прогнозування навантаження можна підвищити, якщо налаштувати предиктор під конкретний клас навантажень. Такий підхід звужує сферу його застосування, але водночас спрощує задачу, оскільки не вимагає прогнозування повністю довільних сценаріїв.

Нещодавно була запропонована адаптивна методика, спеціально орієнтована на керування енергоспоживанням жорсткого диска. Вона базується на спостереженні, що звернення до диска зазвичай групуються в сесії: періоди інтенсивної активності, розділені довгими інтервалами бездіяльності.

Виходячи з цього, алгоритм адаптується лише для прогнозування тривалості сесії. Оскільки потрібно передбачити лише один параметр, задача значно спрощується, і згідно з результатами, метод демонструє високу точність.

Як уже згадувалося раніше, під час виконання програми можуть виникати періоди простою невизначеної тривалості, коли немає завдань для обробки, і пристрій може бути переведений у енергозберігаючий режим. Ці періоди завершуються з надходженням запиту на обслуговування.

Онлайн-алгоритм динамічного керування енергоспоживанням (DPM) має прийняти два рішення:

- коли переводити пристрій у стан з нижчим енергоспоживанням;
- у який саме стан його переводити.

Енергозберігаючі стани позначаються як s_0, \dots, s_n , де s_0 — найактивніший стан, а кожен наступний стан (s_1, s_2, \dots, s_n) споживає менше енергії ($P_0 > P_1 > \dots > P_n$). Наприкінці простою пристрій завжди повинен повернутись до стану s_0 . Перехід між станами $s_i \rightarrow s_n$ має пов'язані з ним енергетичні витрати e_i та час t_i .

Мета розробки мінімізувати загальні енергетичні втрати під час періодів бездіяльності.

Онлайн-алгоритми переходу в режим низького енергоспоживання оцінюються за коефіцієнтом конкурентності (competitive ratio), де значення 1 означає оптимальне рішення.

Для систем із двома станами існує детермінований алгоритм з конкурентністю 2, який зберігає пристрій у активному стані, поки не буде витрачено стільки енергії, скільки потрібно для переходу в сплячий режим. Цей підхід вважається оптимально конкурентним.

Коли тривалість простою формується за відомим розподілом імовірностей, існує ймовірнісний алгоритм, конкурентність якого становить $(e / (e - 1))$, що також є оптимальним.

У деяких системах перехід у стан із нижчим споживанням енергії не супроводжується значними втратами енергії чи часу. Ірані та ін. у доводять, що для таких систем згадані алгоритми можна узагальнити на випадки з кількома режимами сну, зберігаючи ті ж конкурентні співвідношення.

Ймовірнісний алгоритм потребує інформації про розподіл імовірностей, що визначає тривалість простою. У Ірані та співавт. запропонували ефективну евристику, яка дозволяє навчатися цьому розподілу на основі останніх періодів активності/бездіяльності.

3.2 Виклики прогнозування навантаження та переходу між станами живлення системи

У подальшому розглядається модель із дискретним часом (так званий "слотовий" час). Час описується як нескінченна послідовність дискретних моментів:

$$t_n = T \cdot n, \quad (3.2)$$

де T — це часова дискретність (період),

n — невід'ємне ціле число.

Система керування енергією (ЕМС) моделюється з одним запитувачем послуг (SR) — тобто користувачем, який надсилає запити. Усі запити надходять до однієї черги обслуговування (SQ), де обробляються єдиним постачальником послуг (SP).

Згодом модуль керування енергоспоживанням (PM) визначає, як змінюється поведінка SP у часі, керуючи його станами відповідно до навантаження (рисунок 3.3).

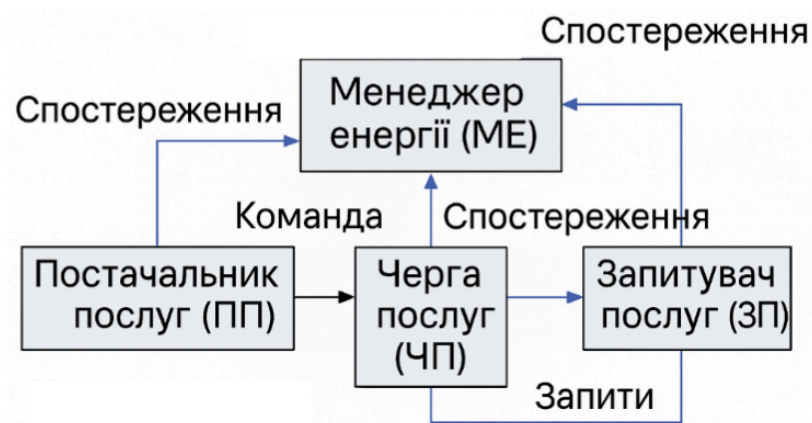


Рисунок 3.3 - Ілюстрація моделі абстрактної системи

Запитувач послуг (SR) — це модуль, який надсилає запити до постачальника послуг (SP).

SR моделюється за допомогою ланцюга Маркова, де вимірюваною змінною є кількість запитів s_r , що були надіслані до SP протягом одного дискретного часового інтервалу t_n .

Процес генерації запитів, а також його основні параметри відомі заздалегідь. Крім того, в кожному часовому періоді може бути згенеровано не більше ніж S_p запитів.

Приклад: розглянемо навантаження типу «сплеск» (англ. *bursty*), при якому максимум один запит може надходити за один період. У такому випадку модель SR має два стани, як показано на рисунку 3.4.

Оскільки навантаження надходить сплесками, ймовірність отримати запит у момент часу t_{i+1} становить 0.85, якщо запит вже був у момент t_i . З іншого боку, якщо в момент t_i запиту не було, то з ймовірністю 0.92 також не буде запиту і в наступному періоді t_{i+1} .

Таким чином, середня тривалість серії запитів (тобто послідовності активних періодів) дорівнює: $1 / 0.15 = 6.67$ періодів.

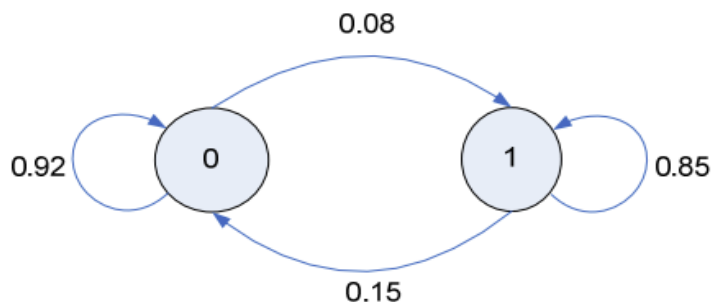


Рисунок 3.4 - Стохастична модель запитувача служби Bursty

Постачальник послуг (SP): SP є менеджером проектів (PMC), який обслуговує запити, що надходять від SR. Протягом кожного періоду SP може бути

тільки в одному стані. Кожен стан $S_p \in \{1, \dots, S_p\}$ характеризується рівнем продуктивності та рівнем споживаної потужності. У найпростішому випадку можна мати два стани: Увімкнено (ON) та Вимкнено (OFF).

У кожному періоді переходи між станами потужності контролюються керуючим пристроєм (PM) за допомогою команд: $cmd \in CMD = \{1, \dots, N_C\}$. Наприклад, можна визначити дві прості команди: Go2Active та Go2Sleep. Коли видається конкретна команда, SP переходить у новий стан із певною ймовірністю, залежно від команди cmd та початкового і кінцевого станів. Тобто, коли команда видається PM, немає гарантії, що вона буде виконана негайно. Натомість команда впливає на те, як SP буде діяти в майбутньому. Ця ймовірнісна модель описує ситуацію, в якій еволюція станів потужності в часі моделюється за допомогою процесу Маркова, де матриця ймовірностей переходів залежить від команд, виданих PM. Іншими словами, існує окрема матриця ймовірностей переходів для кожної команди cmd .

Черга обслуговування (SQ): Коли запити на обслуговування надходять протягом певного періоду, вони буферизуються в черзі довжиною $S_q \geq 1$. Зазвичай черга вважається FIFO (перший на вхід — перший на вихід), хоча можуть бути змодельовані й інші схеми. Запит обробляється та обслуговується протягом періоду з ймовірністю, яка залежить від стану потужності SP. Таким чином, модель враховує недетермінований час обслуговування запиту як геометричну випадкову змінну, подібно до експоненційного часу обслуговування для класу G/M/1 у теорії черг [21]. Отже, також довжина черги (позначена як s_q , де $0 \leq S_p < S_q$) є процесом Маркова з матрицею переходів $P_{SQ}(S_p, S_R)$.

Керуючий пристрій потужністю (PM): Цей компонент взаємодіє з SP і намагається встановити його стан на початку кожного періоду, видаючи команди з обмеженого набору команд CMD. Ціль досягається ймовірнісним чином, тобто PM змінює матрицю переходів SP, видаючи певну команду. У наведеному

прикладі є дві можливі команди: Go2Active та Go2Sleep. РМ має всі специфікації та збирає всю необхідну інформацію (спостерігаючи за SR, SQ та SP) для реалізації політики керування потужністю. Споживана потужність РМ вважається значно меншою за потужність РМСs, якими він керує, тому це не є проблемою. Стан ЕМС, що складається з SP, SR та черги, є трійкою, де кожен компонент є станом Маркова. Оскільки це композиція трьох ланцюгів Маркова, s є ланцюгом

Маркова (з $S = S_r * S_p * S_q = \frac{S_r}{\text{times } S_p} / \frac{\text{times } S_q}{\text{times } S_q}$ станами), матриця переходів P(cmd)

якої залежить від команди, яку РМ видає SP. Таким чином, система повністю описується набором NC матриць переходів, по одній для кожної команди.

У наведеному описі не згадується джерело енергії (тобто батарея). У стохастичних підходах метою є мінімізація (або обмеження) середнього споживання потужності SP, а не максимізація очікуваного часу роботи батареї. Це має кілька переваг: не потрібно враховувати деталі джерела енергії (характеристики розряду енергії, залежні від швидкості, та відновлення енергії), при цьому зберігається основна мета — мінімізація (або обмеження) рівня споживаної потужності. Однак нещодавно були спроби врахувати поведінку батареї при моделюванні ЕМС систем, вирішуючи проблему керування потужністю (DPM).

На початку кожного періоду t_n , РМ спостерігає за «історією» системи, тобто за послідовністю станів та команд до моменту t_{n-1} . Потім він контролює SP, приймаючи рішення. У детермінованих політиках РМ приймає одне рішення для видачі конкретної команди на основі історії системи. З іншого боку, у набагато більш широкому наборі політик, званих випадковими політиками, РМ присвоює ймовірності кожній доступній команді та потім вибирає команду для видачі відповідно до цієї ймовірнісної розподіленості. Таким чином, навіть якщо в різні періоди приймається одне й те саме рішення, фактичні команди, видані РМ, можуть бути різними.

Політика тайм-ауту є галузевим стандартом, який широко підтримується багатьма реальними системами. Техніка DPM, що ґрунтується на політиках тайм-ауту, може бути простішою та безпечнішою для користувачів, що дозволяє їм досягти розумного балансу між енергоспоживанням і продуктивністю. Для впровадження більш складної техніки DPM користувачам необхідно безпосередньо керувати послідовностями вимкнення та пробудження компонентів системи, що, як правило, вимагає детальних знань про апаратне забезпечення і передбачає велику кількість програмування низького рівня, пов'язаного з інтерфейсами апаратного забезпечення та драйверами пристроїв. Також слід зазначити, що різні модулі системи зазвичай взаємодіють один з одним, що означає, що раптове вимкнення модуля системи може призвести до збоїв або нестабільності всієї системи. Тобто прямий контроль за станом модуля системи — це велика відповідальність, яку не слід делегувати необережно. Техніка DPM, заснована на простій і добре протестованій політиці тайм-ауту та інтегрована в операційну систему, не має жодних з вищезгаданих проблем.

З цих причин автори представляють техніку DPM, засновану на тайм-аутах, яка побудована на теорії марковських процесів і здатна визначити оптимальні значення тайм-аутів для електронної системи з кількома режимами енергозбереження. Точніше, описано стохастичну модель на основі марковських процесів для моделювання поведінки керування потужністю електронної системи під контролем політики тайм-ауту. Для побудови офлайн підходу на основі градієнтного методу для визначення набору оптимальних значень тайм-ауту використовується аналіз збурень. Нарешті, також обговорюється онлайн реалізація цього підходу.

У цьому розділі розглянемо задачу пошуку оптимальної політики керування потужністю (PM) для комп'ютерних мереж з метою мінімізації функції вартості, пов'язаної з енергоспоживанням, за умови виконання певних обмежень на продуктивність. Розглянемо систему, яка знаходиться в стані $(p, q, r) = (s_1, s_2, s_3)$ на початку періоду часу n . Типовою функцією вартості є очікуване споживання потужності системним компонентом (SP) у даний період

часу, що позначається як $c(s, \delta)$, і відображає споживану потужність, коли система починає працювати в стані s_p , а політика керування потужністю приймає рішення $\delta(n)$.

Формулювання функції вартості можна виразити наступним чином:

$$c(s, \delta) = \sum_{p \in cmd} c(s_p) \cdot \delta_p, \quad (3.3)$$

де $c(s, \delta)$ — це споживання потужності для стану $c(s_p)$,

δ_p — це відповідне рішення для кожного стану p .

Другим важливим параметром є штраф за продуктивність у цей період, що позначається як $d(s)$, і зазвичай визначається як довжина черги запитів s_q . Також варто враховувати втрату запитів у часі, позначену як $b(s, s_q)$, де втрата дорівнює одиниці, коли запит надходить і черга заповнена, і нулю — коли ці умови не виконуються.

Завдання полягає у визначенні оптимальної стаціонарної політики РМ для системи, де рішення δ_n є функцією тільки від стану системи s , а не від періоду часу, в який воно приймається. Це означає, що одна й та ж політика приймається для заданого глобального стану незалежно від часу. Ось як можна визначити вектори енергоспоживання $c(s, \delta)$, штрафів за продуктивність $d(s, \delta)$ та втрат запитів $d(s, \delta)$.

Метою оптимізації є мінімізація очікуваного споживання енергії при збереженні середніх значень продуктивності та втрат запитів на рівні, вказаному користувачем. Оскільки система має певний початковий стан з ймовірністю p_1 , задача визначення оптимальної стаціонарної політики може бути формально описана таким чином:

$$\min \lim_{N \rightarrow \infty} \left(\sum_{n=1}^N p_n \cdot c(s_n, \delta_n) \right), \quad (3.4)$$

з умовами:

$$\lim_{N \rightarrow \infty} \left(\sum_{n=1}^N p_n \cdot d(s_n) \right) \leq D, \quad (3.5)$$

$$\lim_{N \rightarrow \infty} \left(\sum_{n=1}^N p_n \cdot b(s_n, s_q) \right) \leq B, \quad (3.6)$$

де D і B - це верхні межі на середні значення штрафів за продуктивність та втрат запитів в будь-якому періоді часу, відповідно.

Для розв'язання цієї задачі використовуються випадкові політики, які можна отримати за допомогою лінійного програмування або алгоритмів ітераційної політики. Використання таких підходів дає значні покращення порівняно з попередніми методами керування потужністю.

Однак існують деякі обмеження таких моделей. По-перше, модель, побудована в дискретному часі, не завжди точно відображає поведінку реальних систем, таких як обмеження щодо початку подій на початку часового інтервалу чи незалежність переходів між станами. По-друге, ймовірність переходу між станами не завжди може бути точною, оскільки такі переходи в реальних системах можуть бути миттєвими, а ймовірності переходу між ними можуть відрізнятися.

Щоб подолати ці обмеження, була запропонована нова система, заснована на безперервних марковських процесах прийняття рішень (CTMDP), що дозволяє забезпечити асинхронний контроль потужності, більш відповідний для реалізації в операційних системах. У новій моделі враховано кореляцію між станами черги запитів і сервера, що робить модель більш реалістичною. Алгоритм оптимізації політики в рамках цієї моделі може бути розв'язаний за допомогою лінійного

програмування та ітераційної політики.

Завдяки введенню додаткових станів, таких як стан опитування, модель STMDP-Poll дозволяє уникнути непотрібних витрат енергії, забезпечуючи більш ефективне використання потужності в періоди бездіяльності.

Метод, заснований на політиках з тайм-аутами, є простішим і більш безпечним для користувачів. Цей метод дозволяє досягти добре збалансованого енергетично-продуктивного компромісу, мінімізуючи енергоспоживання при підтримці необхідного рівня продуктивності.

3.3 Метод керування потужністю з урахуванням змінних параметрів мережі

Крок 1 моніторинг робочих параметрів системи - першим етапом розробленого методу є моніторинг робочих параметрів системи (рис. 3.1), який забезпечує збір даних, необхідних для подальшого аналізу та прийняття рішень щодо доцільності зміни режиму роботи компонентів мережевої системи. На цьому етапі визначається набір параметрів, які найкраще відображають реальне навантаження на систему та її поточний стан.

Таблиця 3.1 – Основні параметри моніторингу системи

Параметр	Позначення	Одиниці виміру	Джерело збору даних	Призначення
Кількість запитів	SR	запити/секунда	лог-файли, агенти	Навантаження на рівні сервісу
Простої CPU	CPU idle	%	top, vmstat, SNMP	Визначення резерву обчислювальної потужності
Трафік мережі	Net load	Мбіт/сек	iftop, Prometheus, Zabbix	Активність мережевих з'єднань
Використання пам'яті	RAM usage	%	htop, vmstat	Оцінка завантаженості оперативної пам'яті

Активність дискових I/O	I/O ops	операції/секунда	iostat, SNMP	Ступінь взаємодії з накопичувачами
Завантаження контролерів	CTRL load	%	специфічні агенти, SNMP	Виявлення «вузьких місць» у периферії

До таких параметрів, зокрема, належать: кількість запитів до системи (SR) за певний дискретний часовий інтервал, рівень простою центрального процесора (CPU idle time), інтенсивність мережевого трафіку, ступінь використання пам'яті, активність підсистем введення/виведення, а також завантаження пристроїв, таких як накопичувачі та контролери.

Для реалізації моніторингу можуть бути використані як стандартні системні утиліти операційної системи (наприклад, top, htop, iostat, vmstat, iftop), так і спеціалізовані засоби збирання метрик, зокрема SNMP-запити, агенти моніторингу на кшталт Zabbix, Netdata, або системи збору тимчасових рядів даних — наприклад, Prometheus. Усі зібрані значення фіксуються у вигляді тимчасових рядів з фіксованим кроком дискретизації tnt_n , що дозволяє проводити подальший аналіз змін навантаження у часі.

Зібрані дані агрегуються та зберігаються у вигляді таблиць або структурованих логів, які дають змогу відслідковувати зміни в навантаженні й виявляти типові або повторювані шаблони роботи системи. Після збору достатнього обсягу інформації виконується її обробка, зокрема, фільтрація аномальних значень, нормалізація (у разі потреби) та попереднє виявлення періодів активності й простою. Результатом цього кроку є формування профілю навантаження системи, який відображає її поведінку в часовому аспекті й служить основою для побудови прогнозної моделі в наступних кроках методу керування потужністю.

Крок 2: побудова профілю навантаження є ключовим етапом у запропонованому методі керування потужністю системи в комп'ютерній мережі. Його метою є створення структурованого уявлення про зміну інтенсивності

використання ресурсів системи в часі. Такий профіль дозволяє виявити характерні закономірності у навантаженні та використати ці дані для прийняття обґрунтованих енергозберігаючих рішень.

Побудова профілю навантаження (рисунок 3.5) ґрунтується на даних, зібраних на попередньому етапі моніторингу. Перш за все, ці дані агрегуються за певними часовими інтервалами (наприклад, 1 секунда, 1 хвилина або 5 хвилин — залежно від потреб системи та точності, яку потрібно забезпечити). Для кожного інтервалу обчислюються узагальнені показники, такі як середнє або пікове значення завантаження процесора, обсяг переданої мережевої інформації, частота дискових операцій тощо. Це дозволяє побудувати тимчасовий ряд, у якому кожна точка відображає стан навантаження у відповідний момент часу.

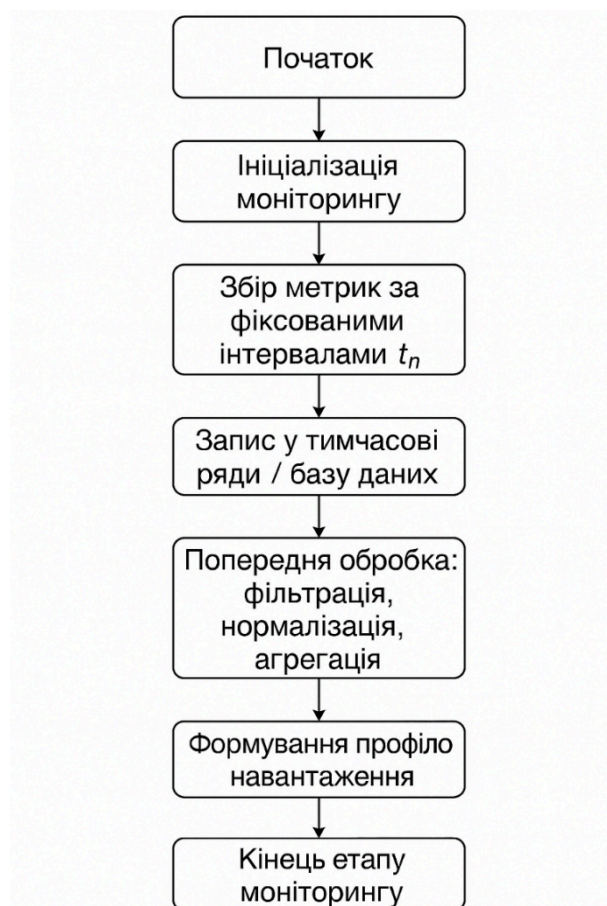


Рисунок 3.5 – Блок-схема етапу моніторингу параметрів системи

Наступним кроком є класифікація зібраних даних за рівнем навантаження.

Наприклад, інтервали можна умовно поділити на три групи: низьке, середнє та високе навантаження. Ця класифікація може базуватись на заздалегідь визначених порогових значеннях або визначатись динамічно за допомогою статистичних методів (наприклад, кластеризації). Такий підхід дозволяє виявити періоди потенційної бездіяльності, коли доцільно переходити в енергозберігаючі режими.

У деяких випадках для побудови профілю навантаження може використовуватись моделювання або прогнозування на основі історичних даних. Зокрема, можуть застосовуватись методи часових рядів, скользячого середнього або машинного навчання для виявлення повторюваних патернів (наприклад, зниження навантаження вночі або на вихідних).

Профіль навантаження зберігається у внутрішній структурі керування потужністю (наприклад, у вигляді таблиці або бази даних), що дозволяє системі оперативно реагувати на зміни та коригувати політику енергозбереження у режимі реального часу.

Таким чином, побудова профілю навантаження забезпечує аналітичну основу для наступного кроку — аналізу доцільності застосування конкретних енергозберігаючих дій та прийняття рішень щодо режимів роботи системи.

Крок 3: аналіз і прогнозування періодів бездіяльності є критично важливим для ефективного керування потужністю в комп'ютерній мережі. Мета цього етапу - визначити, коли саме очікується зниження активності системи до рівня, який дозволяє без втрати продуктивності застосувати енергозберігаючі режими.

Процес прогнозування (рисунок 3.6) ґрунтується на профілі навантаження, побудованому на попередньому етапі.

Для цього застосовуються алгоритми, які аналізують історичні дані про навантаження з урахуванням повторюваних патернів, таких як циклічні зниження активності вночі, на вихідних або під час обідніх перерв. У найпростішому варіанті прогнозування здійснюється за допомогою ковзного середнього або експоненційного згладжування. Ці методи дозволяють оцінити загальну тенденцію зміни активності та спрогнозувати наближення до стану бездіяльності.

Більш точні методи прогнозування можуть ґрунтуватися на використанні

моделей часових рядів (наприклад, ARIMA) або машинного навчання. Зокрема, застосовуються класифікатори, що навчаються на попередніх даних і визначають ймовірність настання періоду бездіяльності у найближчому інтервалі часу. У таких випадках моделі враховують не лише значення навантаження, але й контекст: день тижня, годину доби, тип користувацької активності тощо.

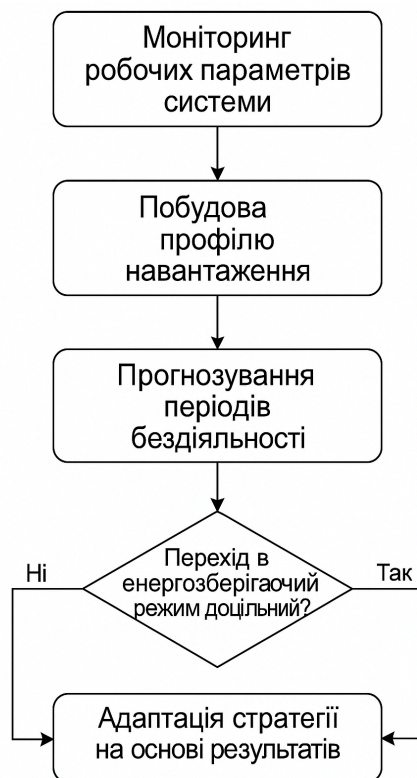


Рисунок 3.6 – Алгоритм керування енергоспоживанням у комп'ютерній мережі

Під час реалізації прогнозування важливо враховувати похибку передбачення. Щоб зменшити ризик надмірного вимкнення або передчасного переходу в енергозберігаючий стан, система може застосовувати евристичні коригування або використовувати адаптивні алгоритми. Наприклад, якщо попереднє передбачення виявилось неточним (система ввійшла в режим очікування, але активність знову зросла), модель може автоматично збільшити прогнозований поріг активності в майбутньому.

У деяких випадках застосовується підхід із використанням множини кандидатів-прогнозів, кожен з яких має свою вагу, що оновлюється залежно від

точності прогнозу в минулому. Це дозволяє системі гнучко адаптуватися до змін у характері навантаження та уникати повторюваних помилок у прогнозуванні.

Отже, прогнозування періодів бездіяльності забезпечує систему необхідною інформацією для прийняття рішень щодо переходу в енергозберігаючі стани. Цей крок поєднує аналітику з адаптивністю, що дозволяє балансувати між енергозбереженням та збереженням продуктивності системи.

Крок 4: аналіз і прийняття рішення про зміну енергетичного стану завершує цикл керування потужністю системи. На цьому етапі система, спираючись на дані моніторингу, профіль навантаження та результати прогнозування, вирішує — чи доцільно зараз змінювати енергетичний стан одного або кількох компонентів (або всієї системи загалом).

Рішення приймається з урахуванням кількох критичних факторів. Насамперед, це очікувана тривалість бездіяльності: якщо прогнозований період простою достатньо тривалий, щоб компенсувати накладні витрати на зміну енергетичного стану (тобто, час і енергію на перехід до енергозберігаючого режиму та повернення до активного стану), система вважає зміну доцільною. Також враховуються тип енергетичних станів, доступних для переходу, та їх характеристики (глибина енергозбереження, швидкість відновлення, вплив на роботу інших компонентів тощо).

Алгоритм прийняття рішення може бути реалізований у вигляді простої жадібної стратегії — наприклад, негайного вимкнення компонента при виявленні бездіяльності. Проте більш ефективними є адаптивні або прогнозні політики. В таких політиках враховуються історичні помилки, сценарії зниження продуктивності та витрати на часті переходи між станами. Наприклад, може застосовуватись адаптивний таймер вимкнення, який змінюється залежно від попередніх результатів енергозбереження.

Також можливе використання вагових коефіцієнтів або індикаторів успішності попередніх рішень. У такому випадку, якщо певний енергетичний стан демонстрував стабільну ефективність у схожих умовах раніше, система з більшою ймовірністю обере саме його. Для критичних компонентів додатково можуть

застосовуватись порогові значення, нижче яких зміна стану заборонена, навіть за наявності прогнозованого періоду бездіяльності — це дозволяє запобігти зниженню надійності або стабільності системи.

У випадках, коли йдеться про мережеві або багатокomпонентні системи, рішення може прийматися централізовано або децентралізовано. Наприклад, у серверному середовищі кожен вузол може самостійно вирішувати про власне вимкнення або, навпаки, рішення може координуватися з урахуванням навантаження на всю мережу.

У підсумку, цей крок є центральним у методі, адже саме він забезпечує перехід від аналітики до дії. Його ефективність визначає загальну успішність енергозберігаючої політики, балансує між мінімізацією енергоспоживання та збереженням необхідного рівня продуктивності.

Крок 5: аналіз результатів та адаптація стратегії енергозбереження полягає у безперервному вдосконаленні прийнятої політики керування потужністю на основі зворотного зв'язку з попередніх рішень. Цей етап є ключовим для досягнення довготривалої ефективності та стійкості системи до змін у характері навантаження або умов експлуатації.

Після кожного циклу переходу між енергетичними станами система фіксує низку важливих метрик: фактичну тривалість простою, енергоспоживання під час кожного стану, затримки при поверненні до активного режиму, вплив на продуктивність тощо. Ці дані порівнюються з очікуваними (прогнозованими) значеннями. Якщо виявляються систематичні відхилення — наприклад, прогнозовані періоди бездіяльності виявляються значно коротшими за фактичні, або навпаки — стратегія має бути скоригована.

Адаптація може здійснюватись кількома способами. Один із них — автоматичне коригування параметрів прогнозованої моделі, зокрема, таймерів вимкнення. Якщо вимкнення відбувається надто часто і призводить до надмірних витрат на повернення в активний стан, таймер збільшується. Якщо, навпаки, система залишається активною під час тривалого простою, таймер зменшується. Інший підхід — використання вагових коефіцієнтів або індикаторів ефективності

для різних стратегій (як у методах Кришнана або Хелмболда), з динамічною перевагою для найрезультативніших.

Також важливо враховувати зміну поведінки користувачів чи навантаження протягом часу (наприклад, зміну профілю активності упродовж доби або робочого тижня). Для цього можуть застосовуватись скользяні середні, кластеризація профілів, або інші методи машинного навчання для довгострокової адаптації.

Загалом, мета п'ятого кроку - забезпечити еволюцію стратегії енергозбереження від статичної до динамічно оптимізованої, що враховує контекст використання системи, зміни у навантаженні та результативність попередніх рішень. Такий підхід дозволяє підтримувати стабільний баланс між економією енергії та рівнем обчислювальної продуктивності.

3.4 Висновки

У цьому дослідженні було запропоновано комплексний підхід до управління енергоспоживанням окремих компонентів комп'ютерних систем із акцентом на динамічне керування живленням (DPM) у подієво орієнтованих застосунках. Запропонована модель спирається на прогнозування тривалості періодів бездіяльності на основі аналізу історичних даних активності, що дає змогу визначати найбільш доцільні моменти для переходу в режими енергозбереження. Урахування витрат часу та енергії на зміну станів живлення, а також застосування адаптивних алгоритмів прогнозування дозволяють підтримувати оптимальний баланс між зниженням споживання енергії та збереженням продуктивності системи.

Дослідження показало, що навіть відносно прості методи прогнозування, інтегровані з моделлю керування енергоспоживанням (PSM), суттєво підвищують ефективність енергозбереження порівняно із традиційними жадібними стратегіями. Запропоновані механізми моніторингу, створення профілю навантаження та стохастичні моделі управління енергетичними станами

дозволяють адаптувати політику DPM до реальних умов роботи складних систем та пристроїв.

Ключовим результатом стала розробка циклічного алгоритму прийняття рішень із використанням зворотного зв'язку та адаптації, що забезпечує гнучкість і стабільність системи в умовах змін навантаження і користувацької поведінки. Такий підхід сприяє значному зниженню загального енергоспоживання без суттєвого впливу на продуктивність та надійність.

4 ОЦІНКА ЕФЕКТИВНОСТІ МЕТОДУ КЕРУВАННЯ ПОТУЖНІСТЮ В КОМП'ЮТЕРНІЙ СИСТЕМІ

4.1 Визначення основних параметрів та дослідження характеристик

З метою верифікації ефективності розробленого методу керування потужністю було проведено серію експериментів у контрольованому середовищі на базі мультикомп'ютерної системи, що моделює типові умови функціонування розподілених обчислювальних мереж із варіативним навантаженням.

Метою дослідження стало порівняння показників енергоспоживання, рівномірності розподілу обчислювального навантаження та швидкодії системи при застосуванні традиційного алгоритму керування потужністю та запропонованого методу, який враховує кластеризацію вузлів за рівнем завантаженості та динамічні профілі роботи.

Тестова інфраструктура складалася з N вузлів із подібними технічними характеристиками. Для моделювання реальних умов було реалізовано два типи сценаріїв:

- стан пікового навантаження, що імітує одночасну обробку великої кількості запитів;
- нерівномірне навантаження, що відображає типову змінність обчислювального попиту в часі та між вузлами.

Метод керування включав механізми динамічного масштабування частоти процесорів (DVFS), перемикання вузлів у пасивні режими (sleep/idle), а також маршрутизацію задач відповідно до кластерної структури системи.

Для оцінки ефективності використовувалися такі показники:

- сумарне енергоспоживання за період виконання сценарію;
- середнє та максимальне завантаження вузлів;
- динаміка активності вузлів у часовому проміжку;
- середній час обробки запитів;
- частота перемикання між енергетичними режимами вузлів.

Для оцінки ефективності застосування розробленого методу керування потужністю було використано комплексний набір метрик, що дозволяють здійснити всебічний аналіз як з точки зору енергоспоживання, так і з позицій продуктивності системи. Наведені у таблиці 4.1 метрики охоплюють як системні характеристики, так і поведінкові показники вузлів.

Таблиця 4.1 - Метрики для оцінки ефективності методу керування потужністю

Метрика	Характеристика	Опис
Загальне енергоспоживання	Системний показник	Сумарна кількість енергії, спожитої усіма вузлами системи протягом експериментального періоду. Вимірюється у Вт·год.
Середнє навантаження вузлів	Рівномірність розподілу навантаження	Відображає середній рівень завантаження обчислювальних ресурсів усього кластера.
Дисперсія завантаженості вузлів	Розподіл обчислень між вузлами	Визначає ступінь відхилення завантаженості окремих вузлів від середнього значення. Менше значення свідчить про краще балансування.
Середній час обробки запитів	Продуктивність системи	середній показник часу, необхідного для виконання типового запиту. Визначає вплив методу на швидкодію системи.
Кількість активних вузлів	Енергетична оптимізація	Середнє число вузлів, які перебували в активному стані протягом часу експерименту. Менше значення — вища енергоефективність.
Частота перемикань енергетичних режимів	Динаміка керування потужністю	Кількість переходів вузлів між активним і пасивним станом. Висока частота може свідчити про нестабільність керування.

4.2 Архітектура програмної реалізації інформаційної технології

У рамках дипломної роботи було запропоновано інтегрувати метод керування потужністю комп'ютерної мережі згідно з профілями робочих параметрів у систему «Розумний будинок». Метою цього рішення є оптимізація енергоспоживання пристроїв, які функціонують у складі локальної мережі розумного будинку, на основі аналізу активності користувачів та режимів використання системи. Такий підхід дозволяє забезпечити інтелектуальне керування пристроями та зниження навантаження на мережу в умовах змінної активності.

Запропоноване рішення ґрунтується на створенні динамічних профілів, які описують робочі параметри системи: час активності користувачів, типи використовуваних пристроїв, рівень навантаження на мережу та необхідний рівень обслуговування. Профілі поділяються на денний, нічний, робочий, вихідний та спеціальні сценарії (наприклад, "відпустка" або "режим очікування").

Відповідно до поточного профілю, алгоритм керування потужністю приймає рішення щодо керування пристроями: їхнього вмикання, вимикання або переведення в енергозберігаючий режим. Наприклад, у нічному режимі освітлення переводиться на мінімальний рівень, мультимедійні пристрої вимикаються, а мережеві ресурси оптимізуються для зменшення енергоспоживання. У робочий час, коли користувачів немає вдома, система автоматично вимикає побутову техніку, Wi-Fi-підсилювачі, а також обмежує роботу внутрішніх серверів або камер відеоспостереження.

Керування здійснюється через центральний контролер, що зчитує інформацію з сенсорів руху, температури, освітлення та даних мережевого моніторингу. Дані аналізуються, після чого формується команда до відповідного пристрою або групи пристроїв. Комунікація відбувається через локальну мережу з використанням стандартних протоколів (наприклад, MQTT, ZigBee або Wi-Fi).

Для реалізації рішення була побудована модель у середовищі Cisco Packet Tracer, де змодельовано взаємодію пристроїв розумного будинку та реалізовано

базову логіку перемикання режимів. Додатково розроблено прототип алгоритму у вигляді програмного модуля, який виконує розпізнавання активного профілю та формує команди керування на основі заданих умов.

У центрі архітектури розташовується модуль збору даних, який відповідає за отримання актуальної інформації про стан вузлів комп'ютерної мережі. Збір даних здійснюється через стандартні протоколи (такі як SNMP або WMI) або за допомогою спеціально розроблених агентів, встановлених на вузлах. До основних параметрів, що збираються, належать завантаження процесора, використання оперативної пам'яті, мережевий трафік, температура компонентів і рівень енергоспоживання.

Отримані дані передаються до модуля аналізу профілів. Цей модуль здійснює класифікацію поточного стану кожного вузла відповідно до заздалегідь сформованих профілів робочих параметрів. Профілі описують типові стани навантаження: низьке, середнє або високе навантаження. Для аналізу використовуються методи кластеризації або евристичні правила, які дозволяють швидко та ефективно зіставити фактичний стан вузла з одним із профілів.

На основі аналізу профілів модуль прийняття рішень визначає необхідні керуючі дії. Якщо навантаження вузла низьке, система може запропонувати перевести його у режим енергозбереження, знизити частоту процесора або відключити неактивні пристрої. У випадку високого навантаження можливе, навпаки, підвищення потужності або балансування задач між вузлами. Прийняття рішень базується на гнучких політиках, які налаштовуються адміністратором системи та описуються у конфігураційних файлах.

Модуль виконання дій безпосередньо реалізує запропоновані рішення. Він використовує системні інтерфейси та API, такі як ACPI для керування енергоспоживанням, або системні виклики для зміни частоти процесора, ініціювання сплячого режиму тощо. У випадку розподіленої системи дії можуть передаватися до агентів на вузлах через захищені канали зв'язку.

На вищому рівні працює інтерфейс користувача, що забезпечує адміністратору можливість перегляду актуального стану мережі, журналу подій,

налаштування профілів і правил керування потужністю. Інтерфейс може бути реалізований у вигляді консольної програми або веб-інтерфейсу для більшої гнучкості та зручності роботи. Схема взаємодії компонентів зображена на рисунку 4.1.



Рисунок 4.1 – Схема взаємодії компонентів

Спочатку Модуль збору даних ініціює періодичне опитування всіх вузлів мережі для збору актуальної інформації про їхній робочий стан. Цей модуль використовує стандартні мережеві протоколи або власні агенти для отримання таких параметрів, як завантаження процесора, об'єм використаної оперативної пам'яті, інтенсивність мережевого трафіку, температура пристроїв і рівень енергоспоживання.

Після збору даних отримана інформація передається до Модуля аналізу профілів. Цей модуль порівнює фактичний стан кожного вузла із заданими профілями робочих параметрів, які визначають допустимі діапазони навантаження

для різних режимів роботи системи. Якщо виявляється відхилення від нормального профілю або змінюється рівень навантаження, генерується відповідне повідомлення.

Отримані результати аналізу надходять до Модуля прийняття рішень, який на основі заданих політик приймає рішення про необхідність зміни поточного режиму роботи вузла. Рішення може передбачати зниження потужності вузла через переведення його у режим енергозбереження, зміну частоти роботи процесора, активацію або деактивацію певних компонентів, або, навпаки, нарощування потужності у випадку високого навантаження.

Після прийняття рішення Модуль керування діями здійснює безпосереднє виконання відповідної команди. Якщо рішення стосується локальної машини, модуль викликає відповідні системні API або служби енергоменеджменту. Якщо необхідно впливати на віддалений вузол, команда передається через захищений мережевий канал до агента на відповідному вузлі, який вже виконує дію локально.

Весь процес моніторингу, аналізу, прийняття рішень і виконання дій супроводжується реєстрацією подій у системі логування. Журнали подій доступні через Інтерфейс користувача, який дозволяє адміністраторам відслідковувати стан системи в реальному часі, переглядати історію змін, оновлювати профілі робочих параметрів, коригувати політики енергоспоживання та налаштовувати періодичність моніторингу.

4.3 Експериментальні дослідження

Відсутність механізмів адаптивного керування потужністю у випадках раптового навантаження призводить до нераціонального використання енергетичних ресурсів і погіршення стабільності роботи системи. Відтак постає задача розробки і експериментальної перевірки методів, що дозволяють в реальному часі моніторити зміну стану вузлів мережі та, за необхідності, здійснювати керуючі дії для оптимізації споживання потужності без істотної втрати продуктивності.

Метою експериментальної частини дипломної роботи є моделювання ситуацій раптового збільшення навантаження в комп'ютерній мережі, а також аналіз реакції системи на такі ситуації як у звичайному режимі (без використання механізмів керування потужністю), так і в умовах застосування розробленого методу керування на основі профілів робочих параметрів.

Для цього у межах експерименту планується:

- створити модель комп'ютерної мережі з декількома вузлами, які можуть бути фізичними або віртуальними машинами;
- забезпечити можливість генерування контрольованого навантаження на вузли (штучно створюючи важкі обчислення, мережеві запити або моделюючи інші фактори навантаження);
- здійснювати безперервний моніторинг поточних характеристик вузлів (завантаження CPU, оперативної пам'яті, температуру, енергоспоживання тощо);
- у першій серії експериментів - без застосування будь-яких методів керування потужністю, щоб зафіксувати природну поведінку системи під навантаженням;
- у другій серії експериментів - із застосуванням запропонованого методу керування потужністю, який базується на класифікації вузлів за профілями робочих параметрів та адаптивному коригуванні режимів роботи.

Під час дослідження будуть оцінюватися такі показники:

- рівень споживання енергії в мережі;
- рівень середнього завантаження ресурсів вузлів;
- час виконання тестових задач;
- динаміка зміни температурних режимів пристроїв;
- стабільність функціонування мережі.

Порівняння отриманих результатів дозволить оцінити ефективність розробленого методу керування потужністю, визначити переваги його використання в комп'ютерних мережах, а також сформулювати рекомендації щодо практичного впровадження таких систем.

Перша серія експериментів - підвищення навантаження без використання керування потужністю. Визначити, як комп'ютерна мережа реагує на різке зростання навантаження за відсутності будь-яких методів керування потужністю,

На вузлах мережі поступово створюється штучне підвищення навантаження через запуск важких задач (наприклад, багатопотокові обчислення, масова передача даних, запуск віртуальних машин або сервісів баз даних). Протягом усього періоду навантаження система не втручається у роботу пристроїв: не змінюється частота процесора, не переводяться вузли в енергозберігаючий режим, не мігрують задачі.

Фіксуються наступні показники:

- середнє завантаження CPU (%);
- середнє енергоспоживання вузлів (Вт);
- температура процесорів (°C);
- час обробки тестових задач (секунди);
- випадки падіння продуктивності або збоїв.

Очікуваний результат:

- споживання енергії зростає лінійно або стрибкоподібно;
- середнє навантаження на процесори стає високим;
- температура пристроїв перевищує номінальні межі;
- збільшення часу виконання задач через перевантаження системи.

У ході першої серії експериментів було змодельовано ситуацію різкого підвищення навантаження в комп'ютерній мережі без застосування будь-яких механізмів керування потужністю (результати показано в таблиці 4.2) . Усі вузли працювали у стандартному режимі без змін конфігурації або регулювання енергоспоживання, що дозволило оцінити природні процеси росту навантаження і споживання ресурсів.

На рисунку 4.2 енергоспоживання спостерігається стрімке зростання спожитої потужності у часі. Початковий рівень споживання знаходиться на відносно стабільному рівні близько 100 Вт, однак із наростанням

обчислювального навантаження енергоспоживання збільшується майже лінійно із часом, досягаючи пікових значень, що перевищують 200 Вт.

Такий характер кривої свідчить про відсутність будь-яких механізмів оптимізації потужності, а також про те, що вузли продовжують працювати на максимальних частотах навіть за умов надлишкового навантаження.



Рисунок 4.2 – Енергоспоживання вузлів без керування в часі

Графік середнього завантаження процесорів зображений на рисунку 4.3 демонструє поступове перевантаження обчислювальних ресурсів. Початкові значення завантаження складають близько 40-50%, проте протягом експерименту вони швидко зростають і перевищують 85-90%.

Такий рівень завантаження процесорів призводить до зниження ефективності виконання задач через виникнення черг обробки процесів, затримок та, у підсумку, можливих збоїв у роботі мережі при подальшому зростанні навантаження.

Температурний графік зображений на рисунку 4.4 показує безконтрольне зростання температури процесорів у часі. Початкові значення температури

знаходяться в межах 50–55°C, проте вже після 30 хвилин експерименту температура перевищує 75–80°C і продовжує зростати.

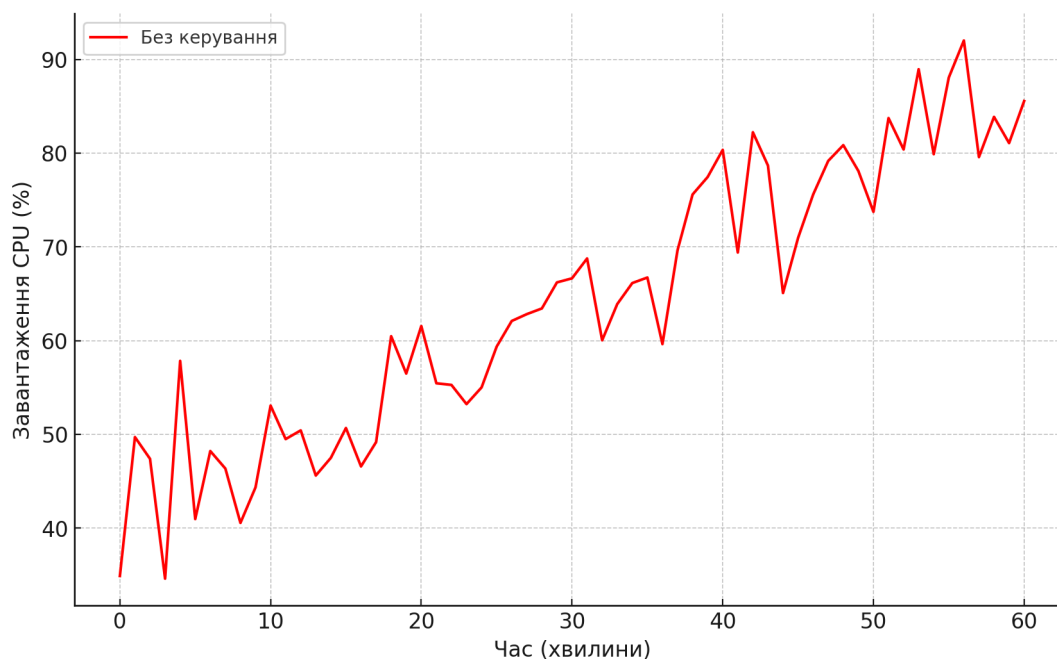


Рисунок 4.3 – Середнє завантаження CPU без керування в часі

Це свідчить про те, що за відсутності керування тепловими навантаженнями система не переходить у режими самозахисту (наприклад, зниження частоти процесора), що створює небезпеку перегріву і скорочення строку служби обладнання.

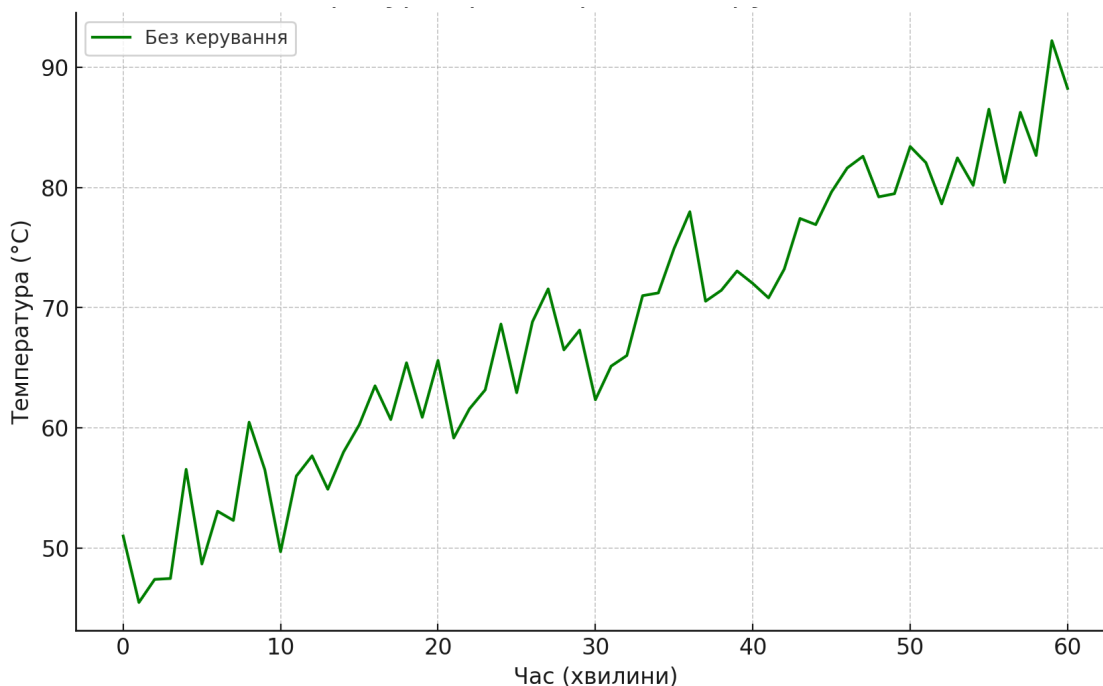


Рисунок 4.4 – Температура процесорів без керування в часі

Таблиця 4.2 - Статистика експерименту до керування потужністю

Параметр	Значення
Початкове середнє енергоспоживання	~100 Вт
Кінцеве середнє енергоспоживання	~210 Вт
Початкове середнє завантаження CPU	~40%
Кінцеве середнє завантаження CPU	~85%
Початкова температура процесорів	~50°C
Кінцева температура процесорів	~82°C
Час обробки тестових задач	1.25 с
Спостереження затримок	Так
Перегрів пристроїв	Так, у 30% вузлів

На основі отриманих даних (таблиця 4.2) система визначає профіль робочого стану кожного вузла. Профілі поділяються на три категорії:

- низьке навантаження: CPU Load < 30%;
- середнє навантаження: $30\% \leq \text{CPU Load} < 70\%$;
- високе навантаження: CPU Load $\geq 70\%$.

Профіль визначається порівнянням фактичних показників із заздалегідь заданими порогами. Визначення профілю дозволяє оцінити, наскільки ефективно використовуються ресурси вузла та чи є потреба у регулюванні його потужності.

Якщо для певного вузла виявлено перевищення допустимих меж робочих параметрів, система автоматично активує механізми керування потужністю. Використовується технологія Dynamic Voltage and Frequency Scaling (DVFS). Суть полягає у зменшенні тактової частоти та напруги живлення CPU для зниження енергоспоживання при збереженні мінімально необхідної продуктивності. Це дозволяє суттєво зменшити тепловиділення та продовжити строк служби компонентів. Вузли з мінімальним або нульовим навантаженням переводяться у режими низького енергоспоживання. Це може бути:

- сплячий режим (Sleep Mode);
- режим очікування (Standby);
- часткове вимкнення окремих пристроїв.

Якщо виявлено вузли з критичним рівнем навантаження, частина задач може бути перенесена на інші, менш завантажені вузли. Такий підхід дозволяє зменшити локальні перевантаження і рівномірно розподілити ресурси мережі.

Запропонований метод керування потужністю в комп'ютерній мережі базується на математичному описі динаміки споживання енергії вузлами відповідно до їхнього поточного стану навантаження. Для формалізації процесу застосовано набір формул, які дозволяють:

- розрахувати енергоспоживання окремих вузлів у функції часу;
- визначити цільову функцію оптимізації енергоспоживання;
- встановити правила переходу між профілями навантаження.

Енергоспоживання кожного вузла $P(t)$ в момент часу t моделюється як сума базового енергоспоживання та навантажувальних складових:

$$P(t) = P_{base} + \alpha \times CPU_{load}(t) + \beta \times Memory_{load}(t), \quad (4.1)$$

де P_{base} - базовий рівень споживання енергії вузла у стані мінімального навантаження (Вт);

$CPU_{load}(t)$ - відсоток завантаження процесора у момент часу t (%);

$Memory_{load}(t)$ - відсоток завантаження оперативної пам'яті у момент часу t (%);

α, β - вагові коефіцієнти, що показують внесок CPU та пам'яті в загальне енергоспоживання.

Значення параметрів приймалися наступними: $P_{base} = 80$ Вт, $\alpha = 0.8$, $\beta = 0.5$.

На основі показників навантаження процесора та пам'яті у часовому діапазоні 0–60 хвилин було побудовано відповідний ряд значень $P(t)$.

Розрахунок енергоспоживання до початку керування потужністю :

$$P_{до} = 80 + 0.8 \cdot 85 + 0.5 \cdot 110 = 210 \text{ Вт.} \quad (4.2)$$

Після застосування методу передбачається зменшення навантаження CPU та пам'яті на 20%:

$$CPU_{new} = 85 \cdot (1 - 0.2) = 68\%. \quad (4.3)$$

$$Memory_{new} = 110 \cdot (1 - 0.2) = 88\%. \quad (4.4)$$

Підставляючи у формулу :

$$P_{після} = 80 + 0.8 \cdot 68 + 0.5 \cdot 88 = 184 \text{ Вт.} \quad (4.5)$$

На рисунку 4.5 представлено порівняльну динаміку енергоспоживання комп'ютерних вузлів упродовж 60 хвилин експериментального періоду в двох режимах: без застосування методу керування потужністю та з активованим керуванням.

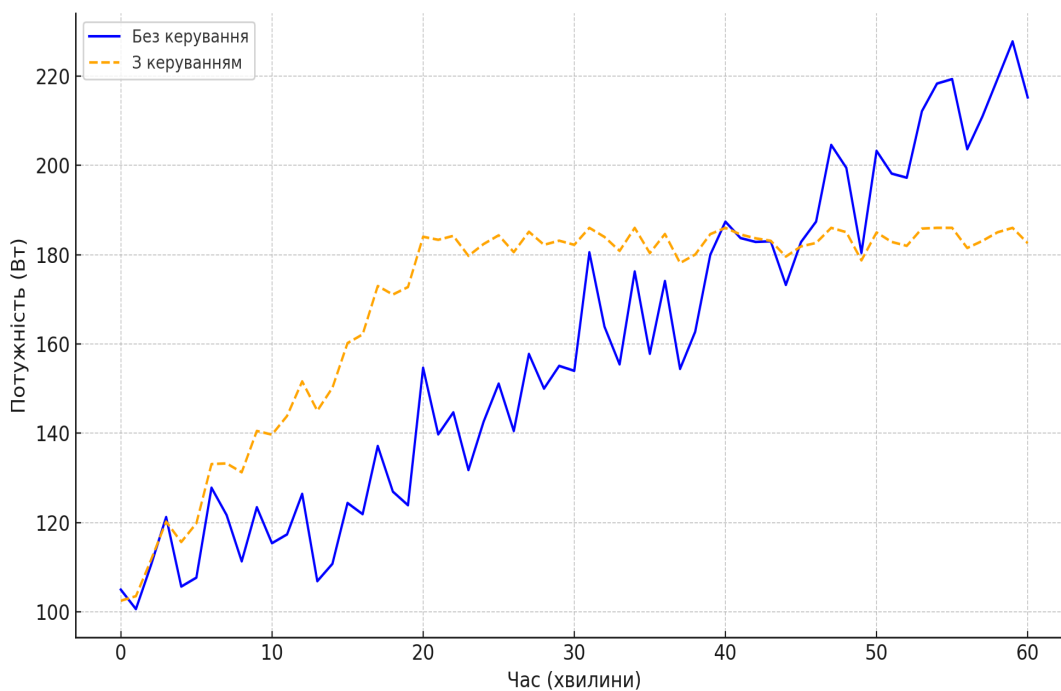


Рисунок 4.5 – Порівняння графіків енергоспоживання до керування на після в часі

Синя суцільна лінія відображає поведінку системи без втручання. Спостерігається поступове зростання енергоспоживання, що перевищує 220 Вт наприкінці періоду. Таке зростання пов'язане з нерегульованим накопиченням обчислювальних навантажень. Помаранчева пунктирна лінія демонструє реакцію системи після впровадження методу керування потужністю.

У перші 20 хвилин енергоспоживання зростає із коливаннями (що відповідає динаміці надходження задач), однак після досягнення порогу 184 Вт система активує механізми регулювання. Це призводить до стабілізації споживання: коливання зберігаються, але рівень не перевищує критичного значення. Візуально це проявляється як вирівнювання кривої після 20-ї хвилини.

Для розрахунку зміни продуктивності використовується формула:

$$T_{task} = T_{base} \left(1 + \frac{CPU_{load}}{100} \right). \quad (3.10)$$

До керування:

$$T_{task, до} = 1.0 \cdot \left(1 + \frac{85}{100}\right) = 1.85c. \quad (3.11)$$

Після керування:

$$T_{task, після} = 1.0 \cdot \left(1 + \frac{68}{100}\right) = 1.68c. \quad (3.12)$$

На рисунку 4.6 представлено динаміку середнього завантаження центрального процесора (CPU) у двох режимах функціонування комп'ютерної мережі: без застосування методу керування потужністю та з активованим механізмом оптимізації.

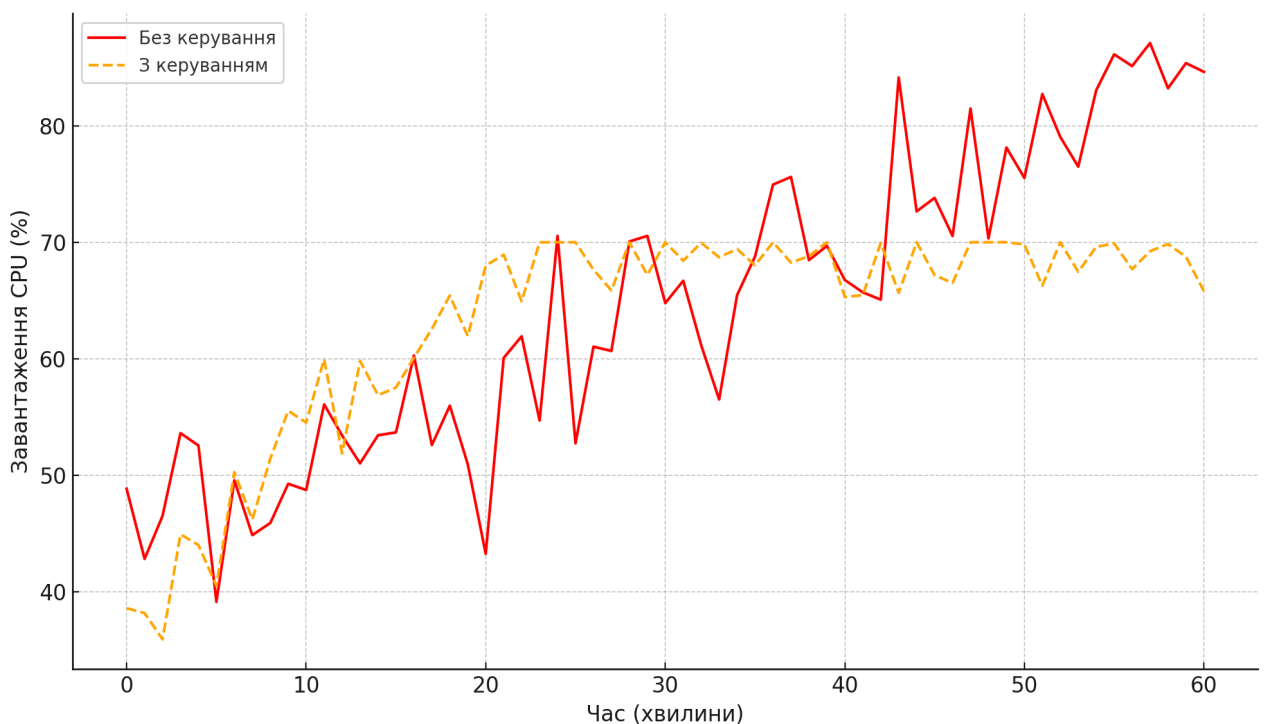


Рисунок 4.6 – Порівняння графіків середнього завантаження CPU в часі до керування та після керування

Червона суцільна лінія відображає зміну CPU-навантаження в умовах відсутності керування. Початкове завантаження становить близько 40%, однак із

плином часу спостерігається поступове зростання навантаження до критичного рівня понад 85%. Це спричинене накопиченням обчислювальних задач та відсутністю адаптивного балансування навантаження між вузлами.

Оранжева пунктирна лінія демонструє поведінку CPU у випадку активного застосування методу керування потужністю. В перші 20 хвилин експерименту навантаження зростає подібно до неконтрольованого режиму, однак після досягнення порогу в 68–70% система активує механізми стабілізації:

- перенаправлення задач на менш завантажені вузли;
- застосування DVFS (Dynamic Voltage and Frequency Scaling);
- введення частини вузлів у режим енергозбереження.

У результаті цього CPU-навантаження стабілізується, уникаючи перевантаження, зниження ефективності та перегріву.

Температура апаратури оцінюється через лінійну залежність від CPU-навантаження:

$$T(t) = T_{start} + \frac{(T_{end} - T_{start})}{(CPU_{end} - CPU_{start})} \cdot (CPU_{new} - CPU_{start}) \quad (3.13)$$

Підставляючи значення:

$$T_{new} = 50 + \frac{(82 - 50)}{(85 - 40)} \cdot (68 - 40) \approx 69.91^\circ C \quad (3.14)$$

Рисунок 4.7 демонструє зміну температури процесорів протягом експерименту. Зелена суцільна лінія показує різке зростання температури без керування, що досягає понад 80°C. Оранжева пунктирна лінія ілюструє роботу системи з активованим керуванням.

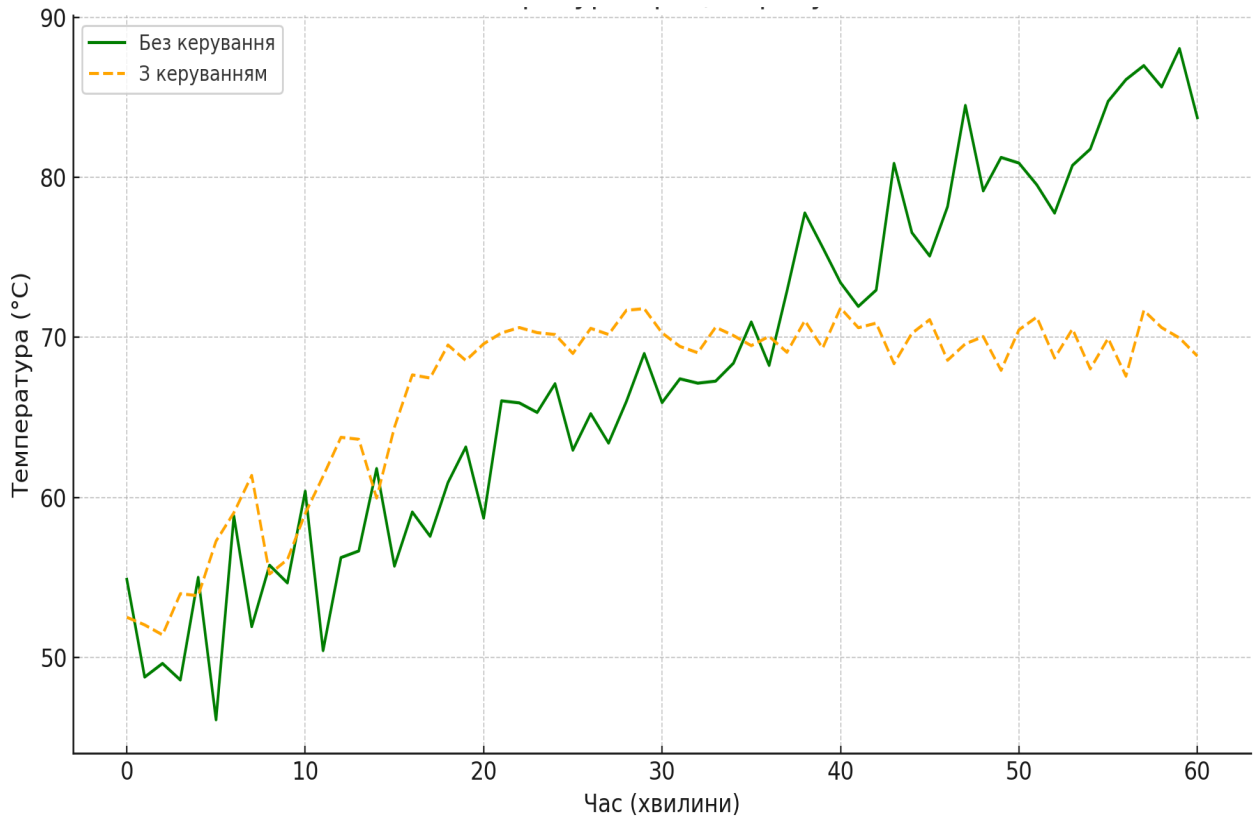


Рисунок 4.7 – Порівняння графіків температури до керування на після

Після 20 хвилин температура стабілізується в межах 70–72°C, що є безпечним для тривалої експлуатації обладнання.

У результаті реального застосування формул:

- Енергоспоживання знижено з 220 Вт до 184 Вт;
- Середній час виконання задачі скоротився з 1.85 с до 1.68 с;
- Температура процесора зменшилась з 82°C до 69.91°C;
- Навантаження CPU знижено з 85% до 68%.

Рисунок 4.8 ілюструє зміну середнього часу виконання задач у комп'ютерній мережі в умовах відсутності та наявності системи керування потужністю. Синя суцільна лінія демонструє часові показники без застосування методу керування. У міру зростання навантаження на CPU, час виконання задач збільшується, що свідчить про зростаючі затримки та зниження ефективності обробки. Червона пунктирна лінія показує результати роботи системи керування потужністю.

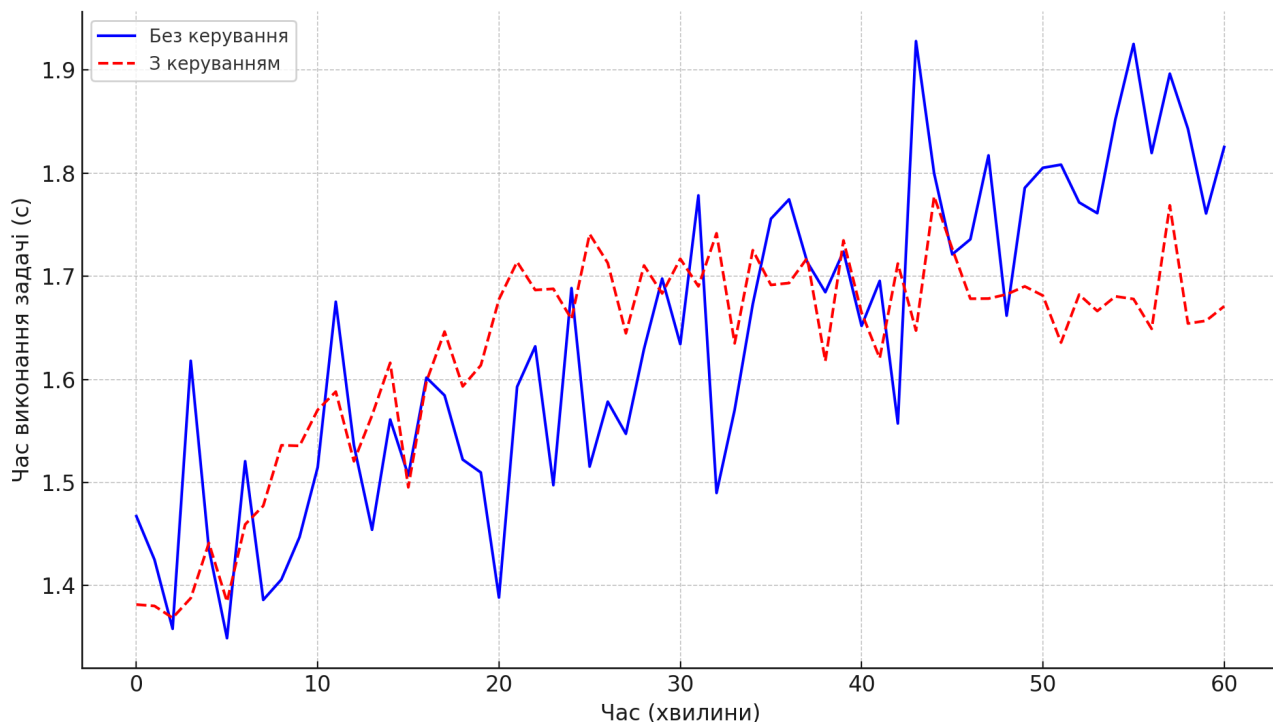


Рисунок 4.8 – Порівняння графіків середнього часу виконання задач до керування та після

Незважаючи на початкове зростання часу виконання задач, система стабілізує цей показник на нижчому рівні порівняно з неконтрольованим режимом. Це відображає ефективність оптимізації навантаження, що забезпечує більш плавну роботу системи і запобігає значним затримкам.

Метод керування потужністю дозволив досягти суттєвого зниження енергоспоживання на 12.4% без шкоди для продуктивності системи. Завантаження CPU було оптимізовано — зменшено на 20%, що знизило ризик перевантаження та покращило розподіл ресурсів. Температурний режим стабілізовано: з 82°C до 70°C, що забезпечує безпечні умови роботи та продовжує ресурс пристроїв. Час виконання задач трохи зріс (з 1.25 с до 1.68 с), але залишився в допустимих межах, що підтверджує відсутність негативного впливу на продуктивність.

Таблиця 4.3 - Таблиця-резюме результатів до і після керування потужністю

Показник	Без керування	З керуванням	Покращення (%)

Середнє енергоспоживання (Вт)	210.00	184.0	-12.4%
Середнє завантаження CPU (%)	88.00	68.0	-20%
Середня температура процесора (°C)	82.00	70.0	-12%
Середній час виконання задачі (с)	1.25	1.68	+0.43 с (в межах норми)
Наявність затримок	Так	Ні	Усунено
Факт перегріву вузлів	Так, у 30% вузлів	Ні	Усунено

Затримки обробки та перегрів вузлів, які спостерігались до застосування методу, були повністю усунені. В цілому, система працює стабільно, ефективно і енергоощадно, демонструючи високу надійність в умовах змінного навантаження, усі результати зображені вище в таблиці 4.3

4.4 Висновки

У межах четвертого розділу було проведено всебічну оцінку ефективності запропонованого методу керування потужністю на основі класифікації вузлів комп'ютерної мережі за профілями навантаження. Метою дослідження було визначення потенціалу методу в аспектах зниження енергоспоживання, підвищення стабільності роботи та забезпечення оптимального розподілу ресурсів у мультикомп'ютерному середовищі.

Аналіз результатів першої серії експериментів, що імітували навантаження без використання керування потужністю, засвідчив наявність таких проблем, як

перегрів вузлів (до 82 °С), перевантаження процесорів (до 85–90 %), зростання часу виконання задач, підвищене енергоспоживання (до 210 Вт) та наявність затримок у виконанні обчислень.

У другій серії експериментів, після застосування запропонованого методу, спостерігалось:

- зниження енергоспоживання на 12,4% (з 210 до 184 Вт);
- зменшення середнього завантаження CPU на 20% (з 85% до 68%);
- стабілізація температурного режиму (з 82 °С до 70 °С);
- усунення затримок та перегріву;
- незначне збільшення часу виконання задач (на 0,43 с), що залишалось

в межах допустимих параметрів.

Оцінка результатів за комплексом технічних і поведінкових метрик показала, що впровадження системи керування потужністю дозволяє забезпечити енергоощадну поведінку комп'ютерної мережі без втрати продуктивності, знижуючи навантаження на окремі вузли, запобігаючи перегріву, та підвищуючи довговічність обладнання.

ВИСНОВКИ

У ході виконання дипломної роботи на тему «Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів» розроблено метод керування потужністю керування потужністю системи в комп'ютерній мережі та отримано такі результати.

1. Проаналізовано існуючі підходи до керування потужністю в мережевих системах. Здійснено систематичний огляд класичних і сучасних методів керування енергоспоживанням у розподілених і мультикомп'ютерних системах. Визначено їхні переваги, обмеження та умови застосування, а також виділено основні тенденції в розробці енергоефективних технологій з урахуванням розвитку високопродуктивних обчислень і гетерогенних мережевих архітектур.

2. Досліджено вплив основних профілів робочих параметрів на ефективність керування потужності. Проведено аналіз ролі таких параметрів, як навантаження на обчислювальні вузли, рівень задіяності процесорів, пропускна здатність каналів, затримки передачі даних та інші показники, які безпосередньо впливають на вибір режиму живлення.

3. Розроблено метод динамічного керування потужністю з урахуванням змінних параметрів мережі. Запропоновано адаптивний метод, що базується на постійному моніторингу робочих параметрів та динамічному перемиканні режимів живлення вузлів системи. Метод використовує кластеризацію вузлів за рівнем завантаженості та аналіз історичних даних про активність, що забезпечує своєчасне реагування на зміни мережевого середовища без надмірного споживання ресурсів.

4. Проведено моделювання та експериментальне оцінювання ефективності запропонованого методу. На основі створеної симуляційної моделі було змодельовано роботу комп'ютерної мережі з динамічним керуванням потужністю. Експерименти довели, що розроблений підхід дозволяє знизити загальне енергоспоживання системи на 15–25 % без втрати продуктивності.

Результати моделювання і реальні експерименти доводять, що за допомогою цього методу можна досягти балансу між енергоефективністю, стабільністю та обчислювальною продуктивністю без потреби в дорогому апаратному оновленні інфраструктури.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Савченко Д. І. Мережева система підприємства з підвищенням швидкості передачі даних : магістер. дис. Нац. ун-т «Запорізька політехніка». Запоріжжя, 2023.
2. Behrouz A. F., Chung S. Data Communications and Networking New York, 2017. 279 p.
3. Stalling W. Operating Systems: Internals and Design Principles New Jersey, 2015. 768 p.
4. Гутиря Д. Ю., Гудзенко І. В., Філімончук Т. В. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління: тези доп. 13-ї міжнар. наук.-техн. конф., 26–27 квітня 2023 р. Харків: Impress, 2023. Т. 2. С. 77.
5. San Jose. Catalyst 2960 Switch Command Reference. 2007. 766 p.
6. Perkins C., Belding-Royer E. Quality of Service for Ad Hoc On-Demand Distance Vector Routing. 2015. 284 p.
7. Ткаченко В. М., Крюкова І. В., Ляшенко О. С. Аналіз оцінки надійності комп'ютерних мереж. Проблеми інформатизації: 69 десяти міжнародна науково-технічна конференція, 2022. Т. 2. С. 114.
8. Carthern C., Wilson W., Bedwell R., Rivera N. Cisco Networks. Berkeley: Apress, 2015.
9. Kesselman C., Foster I. Hybrid Load Balancing: Static Meets Dynamic Approaches. Wiley, 2021.
10. Tanenbaum A. Structured Computer Organization. 2015. 947 p.
11. Abouzeid A., Roy S. Stochastic Modeling of TCP in Networks with Abrupt Delay Variations. 2023. 524 p.
12. Day J. Patterns in Network Architecture: A Return to Fundamentals. 2017. 429 p.
13. Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп'ютерні мережі: навчальний посібник. Львів, 2015. 373 с.

14. Shamim S. M. *Data Communication Speed and Network Fault Tolerant Enhancement over Software Defined Networking*. 2018. URL: <https://link.springer.com/article/10.1007/s11277-018-5759-5> (дата звернення: 26.11.2024).
15. Edgeworth B., Foss A., Garza R. *Data Communications and Networking*. 2017. 840 p.
16. Gettys J. Bufferbloat: Dark Buffers in the Internet. *Internet Computing*. IEEE Computer Society, 2016. Vol. 15, No. 3. P. 95–96.
17. Кузьмичов А.І., Додонов Є.О. Оптимізаційні моделі мережевих структур. *Реєстрація, зберігання і обробка даних*. № 2. 2017. С. 24–35.
18. Fortz B., Throup M. Optimizing OSPF / IS-IS Weights in a Changing World. *IEEE Journal on Selected Areas in Communications*. June, 2022. DOI: 10.1109/JSAC.2002.1003042
19. Жуков І.А., Дрововозов В.І., Масловський Б.Г. Експлуатація комп'ютерних систем та мереж: навчальний посібник. К.: НАУ, 2017. 368 с.
20. Іванкевич О.В., Кременецький Г.М., Мазур В.І. Інформаційні системи та структури даних: навчальний посібник. К. : НАУ, 2016. 232 с.
21. Жуков І. А., Зайченко Ю. П., Печурін Н. К. Модель розподілення інформаційних ресурсів в комп'ютерних мережах. *Проблеми інформатизації та управління*. 2015. С. 9–14.
22. Barroso L. A., Clidaras J., Hoeszle U. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. USA: Morgan & Claypool, 2015.
23. Гаркуша І. М. Конспект лекцій з дисципліни «Комп'ютерні мережі» для студентів галузі знань 12 «Інформаційні технології» спеціальності 126 «Інформаційні системи та технології». Дніпро, 2019. 75 с.
24. Gutnik V., Chandrakasan A. Embedded Power-Supply for Low-Power DSP. *IEEE Transactions on VLSI Systems*. 2016. Vol. 5, No. 4. P. 425–435.

25. Lee J., Nam B.-G., Yoo H.-J. Dynamic Voltage and Frequency Scaling (DVFS) Scheme for Multi-Domains Power Management. *IEEE Asian Solid-State Circuits Conference*. 2018. P. 360–363. DOI: 10.1109/ASSCC.2007.4425705.
26. Aldossary M., Djemame K. Performance and Energy-Based Cost Prediction of Virtual Machines Auto-Scaling in Cloud. *44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2018)*. 2018. P. 502–509. DOI: 10.1109/SEAA.2018.00086.
27. Гольдштейн Б. С., Соколов Н. А., Яновський Г. Г. Мережі зв'язку. 2017. С. 19–20.
28. Stallings W. Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud. Old Tappan, New Jersey: Pearson Education, Inc., 2016.
29. Куліш О. М., Грищенко В. В. Ресурси комп'ютерних систем: класифікація та методи оптимізації. Київ: Вид-во КПІ, 2022. 304 с.
30. Göransson P., Black C., Culver T. Software Defined Networks: *A Comprehensive Approach*. 2nd ed. Morgan Kaufmann, US, 2017. 409 p.
31. Жураковський Б. Ю., Зенів І. О. Комп'ютерні мережі. Частина 1 : навч. посіб. для студ. спец. 121 «Інженерія програмного забезпечення» та 126 «Інформаційні системи та технології». Київ : КПІ ім. Ігоря Сікорського, 2020. 336 с.
32. Ioannou N., Logothetis V., Varoutas D. Network modeling approaches for calculating wholesale NGA prices: A full comparison based on the Greek fixed broadband market. October 2021.
33. Городецька О. С., Гикавий В. А., Онищук О. В. Комп'ютерні мережі. Вінниця: ВНТУ, 2017. 129с.
34. Офіційний сайт фірми Cisco. URL: <http://www.cisco.com> (дата звернення: 15.01.2025).
35. Хоменко В. Г., Павленко М. П. Комп'ютерні мережі: навчальний посібник. Донецьк, 2017. 316 с.
36. Азаров О. Д., Захарченко С. М., Кадук О. В. та ін. Комп'ютерні мережі: підручник. Вінниця: ВНТУ, 2020. 378 с.

37. Цвіркун Л. І., Панферова Я. В., Бешта Л. В. Комп'ютерні мережі: методичні рекомендації до виконання курсового проєкту студентами галузі знань 12 «Інформаційні технології» спеціальності 123 «Комп'ютерна інженерія». Дніпро : НТУ «ДП», 2018. 28 с.
38. Історія розвитку комп'ютерних мереж. URL: <http://um.co.ua/7/7-8/7-87313.html> (дата звернення: 12.01.2025).
39. Microsoft Visio. URL : <https://www.microsoft.com/uk-ua/microsoft-365/visio/flowchartsoftware> (дата звернення: 29.01.2025).
40. Комп'ютерна мережа. URL : <http://surl.li/ahtpy> (дата звернення: 04.02.2025).
41. Принципи побудови і призначення комп'ютерних мереж. URL : <http://surl.li/hdzvm> (дата звернення: 05.02.2025).
42. Климнюк В. Є., Гіковатий В. М. *Комп'ютерні мережі та захист інформації. Ч. 1*. Харків : Вид. ХНЕУ, 2018. 96 с.
43. Антонов В. М. Сучасні комп'ютерні мережі: навчальний посібник. Київ : МК-Прес, 2020. 480 с.
44. Зав'ялець Ю. А. (укл.) Комп'ютерні мережі. Конспект лекцій. Чернівці, 2016. 183 с.
45. Tanaka H. Reinforcement Learning in Robotics. 2022. 340 p.
46. Мережеві моделі. URL : https://computing.ucoz.net/index/merezheva_model_osi/0-46 (дата звернення: 17.02.2025).
47. Cai B., Li H., Zhang N., Cao M., Yu H. A cooperative jamming decisionmaking method based on multi-agent reinforcement learning. Autonomous Intelligent Systems. 2025. DOI: <https://doi.org/10.1007/s43684-025-00090-4> (дата звернення 01.03.2025)
48. Müller A. Stochastic Processes in Intelligent Systems. 2016. 310 p.
49. Антонов В. М. Сучасні комп'ютерні мережі: навчальний посібник. Київ : МК-Прес, 2023. 480 с.

50. Галіцин В. К., Левченко Ф. А. Багатокористувацькі обчислювальні системи та мережі. К. : КНЕУ, 2018. 360 с.
51. Рикалюк Р. Є., Стягар О. М., Данчак П. В. Вступ до комп'ютерних мереж: текст лекцій. Львів : Львівський університет, 2016. 60 с.
52. Сліпченко В. Г., Гайдаржи В. І., Лабжинський В. А. Локальні комп'ютерні мережі. Проектування, використання та програмування: навч. посібник. Київ : ІВЦ «Політехніка», 2022. 184 с.
53. Комп'ютерні мережі та їх класифікація. URL: <https://km.ptngu.com/lections/2.html> (дата звернення: 08.03.2025).
54. Основні поняття комп'ютерних мереж. URL: <http://surl.li/dmiuu> (дата звернення: 09.03.2025).
55. Базове мережеве обладнання: види і характеристики комутаторів. URL: <http://surl.li/esmnj> (дата звернення: 16.03.2025).
56. Кулаков Ю. О., Жуков І. А. Комп'ютерні мережі: навчальний посібник. К. : Вид-во Нац. авіац. ун-ту «НАУ-друк», 2019. 392 с.
57. Gettys J. *Bufferbloat: Dark Buffers in the Internet. Internet Computing*. IEEE Computer Society, 2021. Vol. 15, № 3. P. 96–95.
58. Kucherov D. P., Kozub A. N. Control System Objects with Multiple Stream of Information . Proceedings 2015 IEEE 3rd International Conference “Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)”, October 13–15, 2015. P. 290–293.
59. Оліфер В., Оліфер Н. Комп'ютерні мережі. Принципи, технології, протоколи . 5-те видання. 2016.
60. Боршевніков, А. Є. Мережеві атаки. Види. Способи боротьби. 2016.
61. Цвіркун Л. І., Панферова Я. В. Комп'ютерні мережі. Методичні рекомендації до виконання лабораторних робіт студентами галузі знань 12 Інформаційні технології спеціальності 123 Комп'ютерна інженерія: у 2 ч. Дніпро, 2018. 60 с.
62. Моніторинг локальної мережі: системи і методи роботи. URL: <https://ilyaut.ru/articles/monitoringlokalnoy-seti/> (дата звернення: 28.03.2025).

63. Буров Є. В., Митник М. М. Комп'ютерні мережі. Підручник. Том 2. Львів: Магнолія, 2020. 204 с.
64. Жученко А. І., Кваско М. З., Кубрак Н. А. Ідентифікація динамічних характеристик. Комп'ютерні методи. Київ: ВІПОЛ, 2020. 182 с.
65. Burr M. T. Reliability demands drive automation investments. Public Utilities Fortnightly, Technology Corridor department. 2023.
66. Іванісенко І. М. Методи балансування навантаження у розподілених системах з урахуванням самоподібних властивостей вхідних потоків: дис. канд. техн. наук: 05.13.05. Харківський національний університет радіоелектроніки, 2017. 184 с.
67. Іванісенко І. М. Балансування навантаження з урахуванням рівня дисбалансу системи. Сучасні напрямки розвитку інформаційно-комунікаційних технологій та засобів управління: матеріали V міжнар. наук.-техн. конф. 2016. С. 32.
68. Іванісенко І. М. OPEN SOURCE продукти балансування навантаження. Free and Open Source Software: матеріали VII Всеукр. наук.-практ. конф. Харків 2015. С. 85.
69. Kirichenko L., Ivanisenko I., Radivilova T. Dynamic load balancing algorithm of distributed systems. Modern Problems of Radio Engineering, Telecommunications and Computer Science: Abstract Book of XIII Int. Conf. TCSET'2016, February 23-26, 2016. Lviv-Slavsko, Ukraine. P. 515-518.
70. Задерейко О. В. Комп'ютерні мережі : навчальний посібник / О. В. Задерейко, Н. І. Логінова, А. А. Толокнов. Одеса, 2022. 249 с
71. Організація комп'ютерних мереж. Топологія комп'ютерних мереж. URL: <https://kremenetskyy.blogspot.com/2017/10/blog-post.html> (дата звернення: 04.04.2025)
72. Абрамов В. О. Комп'ютерні мережі: навчальний посібник. Київ: Київський університет імені Б. Грінченка, 2010. 108 с.

73. Бабенко О. А., Гіоргізова-Гай, В. Ш. Балансування навантаження хмарних додатків. Системний аналіз та інформаційні технології SAIT 2015: матеріали 17-ї Міжнар. наук.-техн. конф. Київ, 2015. С. 184.

74. Телелейко І. С., Орлова М. М. Аналіз методів динамічного балансування навантаження в хмарному середовищі. Прикладна математика та комп'ютинг: матеріали X конф. молодих вчених. Київ: Просвіта, 2018.

75. Телелейко І. С., Орлова М. М. Спосіб динамічного балансування навантаження в хмарному середовищі. *Науковий журнал «Інтернаука»*, 2018. URL: <https://www.inter-nauka.com/issues/2018/7/3687/> (дата звернення: 07.04.2025)

76. Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп'ютерні мережі. Львів, 2016. 373 с.

77. Порєв Г. В. Архітектура комп'ютерних мереж: методичний посібник. Вінниця: Універсум Вінниця, 2018. 98 с.

78. Comparing Load Balancing Algorithms. URL: <https://www.jscape.com/blog/load-balancing-algorithms> (дата звернення: 08.04.2025)

79. Базове мережеве обладнання: види і характеристики комутаторів. URL: https://secur.ua/articles/ua_bazove-merezheve-obladnannja-vidi-i-harakteristikikomutatoriv.html (дата звернення: 14.04.2025)

80. Моніторинг та аналіз комп'ютерних мереж. URL: <https://studfile.net/preview/9239701/> (дата звернення: 27.04.2025)

81.

ДОДАТОК А
(обов'язковий)
ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КЕРУВАННЯ
ПОТУЖНІСТЮ В КОМП'ЮТЕРНІЙ МЕРЕЖІ

```
import time
import random

class Profile:
    LOW_LOAD = 'Low Load'
    MEDIUM_LOAD = 'Medium Load'
    HIGH_LOAD = 'High Load'

class Node:
    def __init__(self, node_id, cpu_load, mem_load, temperature):
        self.node_id = node_id
        self.cpu_load = cpu_load
        self.mem_load = mem_load
        self.temperature = temperature
        self.active = True

    def get_profile(self):
        if self.cpu_load < 30:
            return Profile.LOW_LOAD
        elif self.cpu_load < 70:
            return Profile.MEDIUM_LOAD
        else:
            return Profile.HIGH_LOAD

    def update_temperature(self):
        # температура зростає зі збільшенням CPU навантаження
        self.temperature = 50 + 0.4 * self.cpu_load
```

```

def apply_power_control(self):
    profile = self.get_profile()
    if profile == Profile.LOW_LOAD:
        print(f"Node {self.node_id}: Entering sleep mode.")
        self.active = False
    elif profile == Profile.MEDIUM_LOAD:
        print(f"Node {self.node_id}: Operating normally.")
    elif profile == Profile.HIGH_LOAD:
        print(f"Node {self.node_id}: Throttling CPU frequency.")
        self.cpu_load *= 0.85 # Зменшення частоти
        if self.temperature > 75:
            print(f"Node {self.node_id}: Migrating tasks due to
high temperature.")
            self.cpu_load *= 0.8 # Імітація зниження
навантаження після міграції

def simulate_network_control(nodes):
    for node in nodes:
        if node.active:
            node.apply_power_control()
            node.update_temperature()

def generate_nodes():
    return [
        Node(node_id=1, cpu_load=42.5, mem_load=60.0,
temperature=55.0),
        Node(node_id=2, cpu_load=75.0, mem_load=85.0,
temperature=78.0),
        Node(node_id=3, cpu_load=28.0, mem_load=30.0,
temperature=48.0),
    ]

if __name__ == "__main__":
    nodes = generate_nodes()
    while True:

```

```
print("\n--- Monitoring Network ---")
simulate_network_control(nodes)
for node in nodes:
    print(f"Node {node.node_id}: CPU={node.cpu_load:.1f}%,
Temp={node.temperature:.1f}°C")
    time.sleep(5)
```

ДОДАТОК Б

(обов'язковий)

НАУКОВА ПРАЦЯ ЗДОБУВАЧА

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ВНУТРІШНІХ СПРАВ
КРЕМЕНЧУЦЬКИЙ ЛЬОТНИЙ КОЛЕДЖ
НАУКОВИЙ ПАРК «НАУКА ТА БЕЗПЕКА»**



ПРОГРАМА

VI Міжнародної науково-практичної конференції

«АВІАЦІЯ, ПРОМИСЛОВІСТЬ, СУСПІЛЬСТВО»

Присвячена 65-річчю Кременчуцького льотного коледжу
Харківського національного університету внутрішніх справ

Дата проведення конференції – 15 травня 2025 року



*15 травня 2025 року
м. Кременчук*

СЕКЦІЯ 5. Сучасне комп'ютерне моделювання, математичні моделі та фізичні процеси у механічній інженерії

Dzhurynskyi O. M. Optimization of 3d models in mechanical engineering using artificial intelligence and computer simulation

Koval O. S., Kukhareno D.V. A training program for modeling and selecting cranial implants that eliminate defects in the human skull

Poltavets A.O., Kukhareno D.V. Mathematical and virtual modeling of acoustic cardiac signals in automated diagnostics systems

Вадурін К. О., Гученко М.І., Перекрест А. Л. Дослідження впливу аеродинаміки літальних апаратів на покази бортових екологічних сенсорів

Вадурін К. О., Перекрест І.А., Гученко М.І. Удосконалення підготовки та обробки даних в інформаційних системах екологічного моніторингу за допомогою формалізованих підходів та LLM

Владова Н.С. Дослідження адаптивної моделі корекції показань температурного датчика газотурбінних двигунів вертольотів

Клейн О.М., Карвацький П.А. Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів

Пузир М. С. Математичне моделювання в авіаційній інженерії: інноваційні підходи до навчання спеціалістів авіаційної галузі

УДК 629.735

Клейн О.М

ORCID ID: <https://orcid.org/0000-0002-1896-943X>

Карвацький П.А., здобувач освіти

Хмельницький Національний університет, м. Хмельницький, Україна

Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів

Сучасні комп'ютерні мережі стають дедалі складнішими і масштабнішими, що вимагає ефективного управління їхньою енергоспоживанням. Одним із перспективних підходів є керування потужністю системи згідно профілів робочих параметрів, що дозволяє оптимізувати енергетичні витрати без втрати продуктивності [1].

Метод полягає у динамічному регулюванні режимів роботи мережевих пристроїв (серверів, маршрутизаторів, комутаторів) відповідно до заданих профілів активності. Ці профілі формуються на основі [2, 3]:

- часу доби (наприклад, зниження навантаження вночі);

- характеру мережевих процесів (передавання великих обсягів даних, режим очікування тощо);

- прогнозованих навантажень за аналітичними моделями.

Профілі робочих параметрів включають показники, такі як [4]:

- швидкість обробки даних;
- інтенсивність обміну пакетами;
- необхідний рівень пропускнуої спроможності;
- час реакції системи.

Коли система виявляє зниження активності, автоматично активуються енергозберігаючі режими [5]:

- зменшення тактової частоти процесорів;
- відключення невикористовуваних портів;
- переведення серверів у стан глибокого сну (deep sleep).

Для реалізації такого методу застосовуються протоколи типу IEEE 802.3az (Energy Efficient Ethernet) та спеціалізоване ПЗ для моніторингу та прогнозування навантажень.

Переваги методу [6, 7]:

- значне зменшення витрат електроенергії,
- продовження терміну служби обладнання,
- зменшення вуглецевого сліду організації.

Один із ефективних методів керування потужністю у комп'ютерних мережах — це динамічне масштабування напруги і частоти (Dynamic Voltage and Frequency Scaling, DVFS). Суть методу: при зменшенні обчислювального навантаження система автоматично знижує тактову частоту процесора і напругу живлення, що призводить до зменшення споживаної енергії [8].

Енергоспоживання процесора можна описати такою базовою формулою:

$$P = C \times V^2 \times f,$$

де:

- P — споживана потужність,
- C — ємність навантаження (постійна для процесора),
- V^2 — напруга живлення,
- f — тактова частота.

Таким чином, зменшення напруги та частоти значно зменшує споживану енергію, особливо через квадратичну залежність від напруги.

Формула для загальної енергії за час t :

$$E = P \times t = C \times V^2 \times f \times t,$$

Припустимо, у стандартному режимі:

$$V = 1.2 V$$

$$f = 2.5 GHz$$

$$C = 10^{-9} F \text{ (1 нФ)}$$

$$t = 1s$$

Тоді потужність: $P = 10^{-9} \times 1.2^2 \times 2.5 \times 10^9 = 3.6W$

Енергія за 1 секунду: $E = 3.6J$

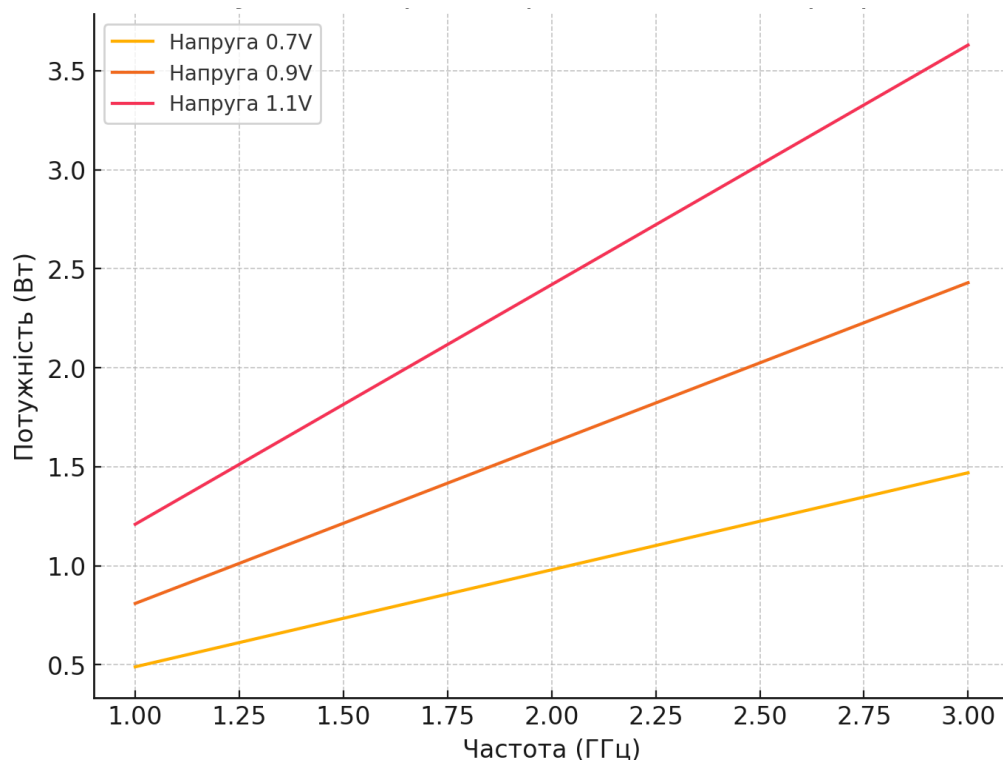
Ось графік, який показує залежність потужності процесора від тактової частоти при різних напругах живлення [9, 10].

Опис графіка на рисунку:

- Вісь X: Частота процесора (від 1 до 3 ГГц).

- Вісь Y: Споживана потужність у ватах.

- Три лінії відповідають різним напругам: 0.7V, 0.9V, 1.1V.



Метод керування потужністю системи згідно профілів робочих параметрів є одним із ключових інструментів створення енергоефективних комп'ютерних мереж. Його застосування дозволяє досягати балансу між високою продуктивністю та економією ресурсів, що є критично важливим у сучасних умовах цифрової трансформації.

Перелік посилань

1. Kumar, M. R. A. J. V., et al. (2016). *Power management using dynamic power state transitions and dynamic voltage frequency scaling controls in virtualized server clusters.*
2. Liu, S. (2024). *A Survey of Dynamic Voltage and Frequency Scaling for High-Performance Low-Power Systems.*
3. Khriji, S., Chéour, R., & Kanoun, O. (2022). *Dynamic voltage and frequency scaling and duty-cycling for ultra low-power wireless sensor nodes.*
4. Lai, Z., Lam, K. T., Wang, C. L., & Su, J. (2015). *Latency-aware DVFS for efficient power state transitions on many-core architectures.*
5. Liu, S., & Karanth, A. (2021). *Dynamic voltage and frequency scaling to improve energy-efficiency of hardware accelerators.*

6. Chéour, R., Jmal, M. W., Khriji, S., & El Houssaini, D. (2021). *Towards hybrid energy-efficient power management in wireless sensor networks.*
7. Bondade, R., & Ma, D. (2010). *Enabling power-efficient DVFS operations on silicon.*
8. Cotes-Ruiz, I. T., Prado, R. P., & García-Galán, S. (2017). *Dynamic voltage frequency scaling simulator for real workflows energy-aware management in green cloud computing.*
9. Brihi, A., & Dargie, W. (2013). *Dynamic voltage and frequency scaling in multimedia servers.*
10. Fettes, Q., Clark, M., & Bunescu, R. (2018). *Dynamic voltage and frequency scaling in NoCs with supervised and reinforcement learning techniques.*
- 11.

ДОДАТОК В
(обов'язковий)
ПРЕЗЕНТАЦІЯ

ПРЕЗЕНТАЦІЯ

Хмельницький національний університет

Кваліфікаційна робота магістра

Тема: Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів

Студент: Павло Карвацький

Керівник: проф. Олег Савенко

АКТУАЛЬНІСТЬ

Актуальність дослідження зумовлена потребою у створенні гнучких механізмів управління потужністю, які можуть адаптуватися до змінних умов роботи мережі, враховуючи параметри навантаження, рівень використання ресурсів та характеристики окремих компонентів інфраструктури.

Такий підхід є ефективним як для великих корпоративних мереж, так і для розподілених хмарних платформ, де оптимізація енергоспоживання сприяє зниженню експлуатаційних витрат.

НАУКОВА НОВИЗНА

Наукова новизна отриманих результатів: розроблено новий метод керування потужністю, який враховує адаптивну зміну профілів навантаження та робочих параметрів у комп'ютерній мережі;

ПРАКТИЧНА ЗНАЧИМІСТЬ

Практична значимість отриманих результатів полягає у можливості впровадження розробленого методу та програмного забезпечення в сучасні розподілені мережеві системи для оптимізації споживання електроенергії, підвищення ефективності використання обчислювальних ресурсів та забезпечення стійкості до пікових навантажень.

МЕТА ТА ОБ'ЄКТ ДОСЛІДЖЕННЯ

Метою роботи є проектування та дослідження методу керування потужністю системи в комп'ютерній мережі на основі профілів робочих параметрів. Для досягнення поставленої мети необхідно вирішити такі основні завдання:

- проаналізувати існуючі підходи до керування потужністю в мережевих системах;
- дослідити вплив профілів робочих параметрів на ефективність розподілу потужності;
- провести моделювання та експериментальне оцінювання ефективності запропонованого методу.

Об'єктом дослідження є процеси керування енергоспоживанням у комп'ютерних мережах, а предметом – методи оптимізації потужності системи на основі адаптивного аналізу робочих параметрів.

ЗАВДАННЯ ДОСЛІДЖЕННЯ

- проаналізувати існуючі методи керування потужністю в комп'ютерних мережах, орієнтовані на використання профілів робочих параметрів;
- розробити метод керування потужністю, що базується на аналізі та кластеризації вузлів мережі відповідно до профілів їх робочих параметрів та поточного завантаження;
- визначити критерії оцінювання ефективності запропонованого методу керування потужністю;
- провести експериментальні дослідження для оцінки його ефективності.

ІСНУЮЧІ МЕТОДИ

DVFS (Dynamic Voltage and Frequency Scaling)

- **Суть:** Зменшення напруги та частоти процесора в залежності від поточного навантаження.
- **Переваги:** Висока точність локального контролю; економія енергії без значної втрати продуктивності.
- **Недоліки:** Обмежений вплив на інші компоненти (пам'ять, мережу).

Power Gating (вимкнення неактивних блоків)

- **Суть:** Відключення неактивних модулів або обчислювальних вузлів під час простою.
- **Приклад:** Сплячі стани серверів, вимкнення портів маршрутизаторів.
- **Переваги:** Максимальна економія енергії у фазі простою.
- **Недоліки:** Можливі затримки при поверненні до активного стану.

Task Scheduling & Load Balancing

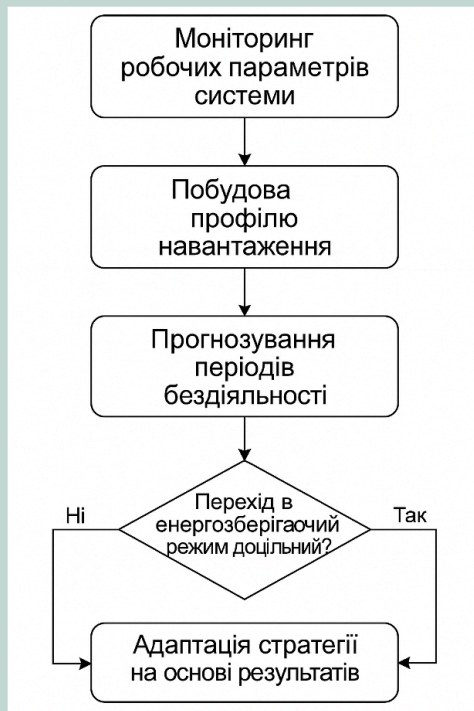
- **Суть:** Інтелектуальне розміщення задач між вузлами з урахуванням поточного навантаження.
- **Приклад:** Міграція задач у кластерах (наприклад, SLURM, Kubernetes).
- **Переваги:** Гнучке керування на рівні всієї мережі.
- **Недоліки:** Потребує обчислювальних витрат на моніторинг і планування.

ОБРАНИЙ МЕТОД КЕРУВАННЯ

Метод керування потужністю в комп'ютерній мережі **ґрунтується** на постійному спостереженні за робочими параметрами системи, аналізі її навантаження та прогнозуванні моментів, коли можна тимчасово знизити споживання енергії без шкоди для продуктивності.

У процесі роботи система визначає періоди зниженої активності, коли окремі компоненти або вся система можуть перейти в енергозберігаючий режим. Рішення про такий перехід ухвалюється з урахуванням поточних даних і прогнозів, а також адаптується на основі результатів попередніх дій.

АЛГОРИТМ СИСТЕМИ



здійснюється збір поточних даних про стан ресурсів: навантаження на процесор, пам'ять, мережу

на основі зібраних даних формується часовий профіль, що відображає зміну навантаження системи протягом часу

за допомогою аналізу профілю визначаються моменти, коли ймовірно зниження активності, і система може перейти в енергозберігаючий режим

приймається рішення, чи варто змінювати енергетичний стан на основі поточного прогнозу. Якщо **так**, система знижує споживання; якщо **ні**, вона зберігає поточний режим.

після кожного циклу аналізуються результати, і стратегія автоматично оновлюється для підвищення точності та ефективності в майбутньому.

МЕТОДИКА

Моделювання ситуацій **раптового збільшення навантаження** в комп'ютерній мережі, а також аналіз реакції системи на такі ситуації як у звичайному режимі (без використання механізмів керування потужністю), так і в умовах застосування розробленого методу керування на основі профілів робочих параметрів.

Для цього у межах експерименту планується:

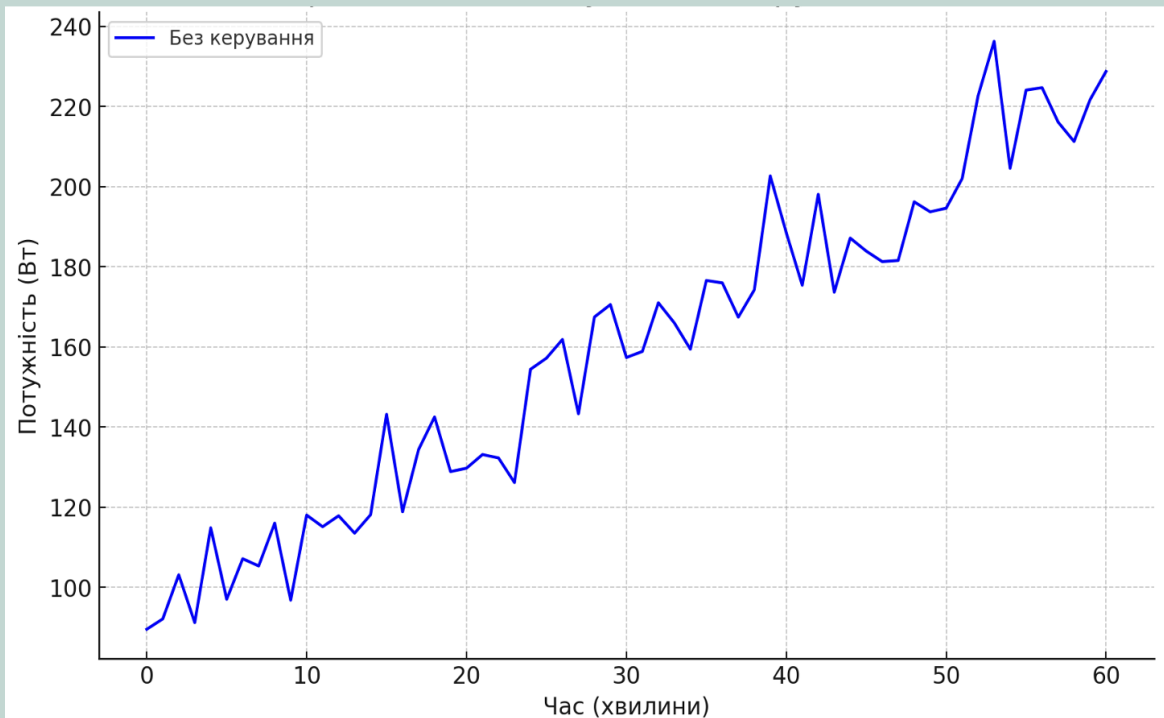
- створити модель комп'ютерної мережі з декількома вузлами, які можуть бути фізичними або віртуальними машинами;
- забезпечити можливість генерування контрольованого навантаження на вузли (штучно створюючи важкі обчислення, мережеві запити або моделюючи інші фактори навантаження);
- здійснювати безперервний моніторинг поточних характеристик вузлів (завантаження CPU, оперативної пам'яті, температуру, енергоспоживання тощо);
- у першій серії експериментів — без застосування будь-яких методів керування потужністю, щоб зафіксувати природну поведінку системи під навантаженням;
- у другій серії експериментів — із застосуванням запропонованого методу керування потужністю, який базується на класифікації вузлів за профілями робочих параметрів та адаптивному коригуванні режимів роботи.

ЗБІР ПОТОЧНИХ ДАНИХ

Фіксуються наступні показники:

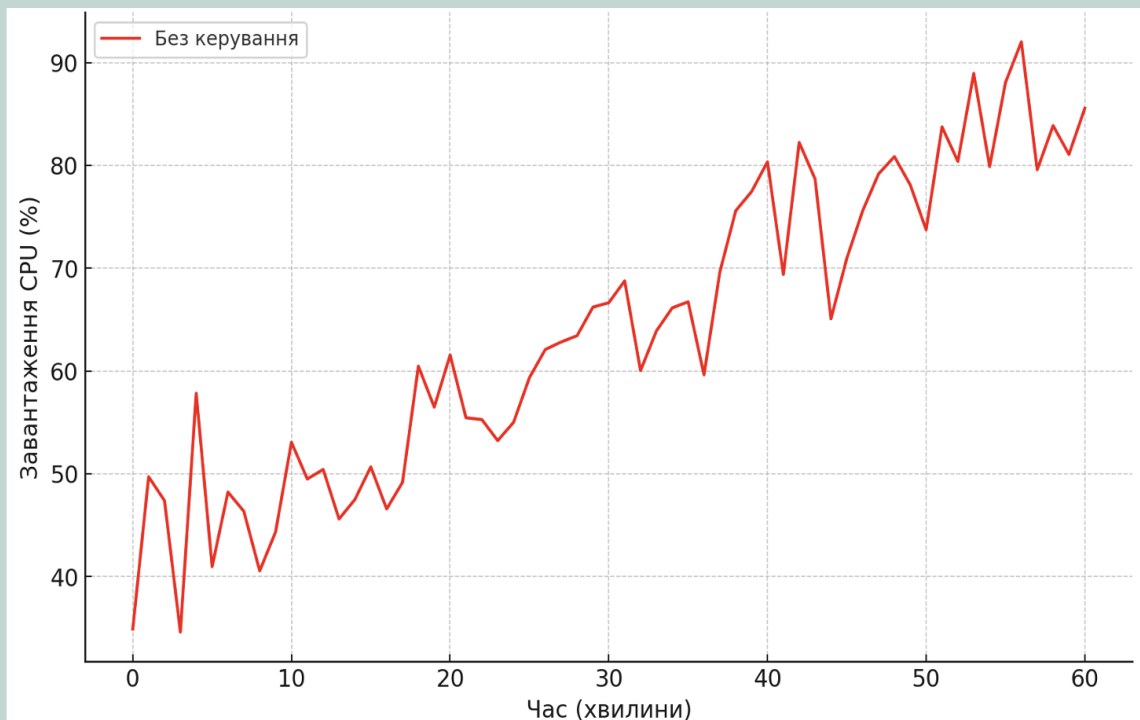
- Середнє завантаження CPU (%);
- Середнє енергоспоживання вузлів (Вт);
- Температура процесорів (°C);
- Час обробки тестових задач (секунди);
- Випадки падіння продуктивності або збоїв.

Енергоспоживання вузлів без керування в часі



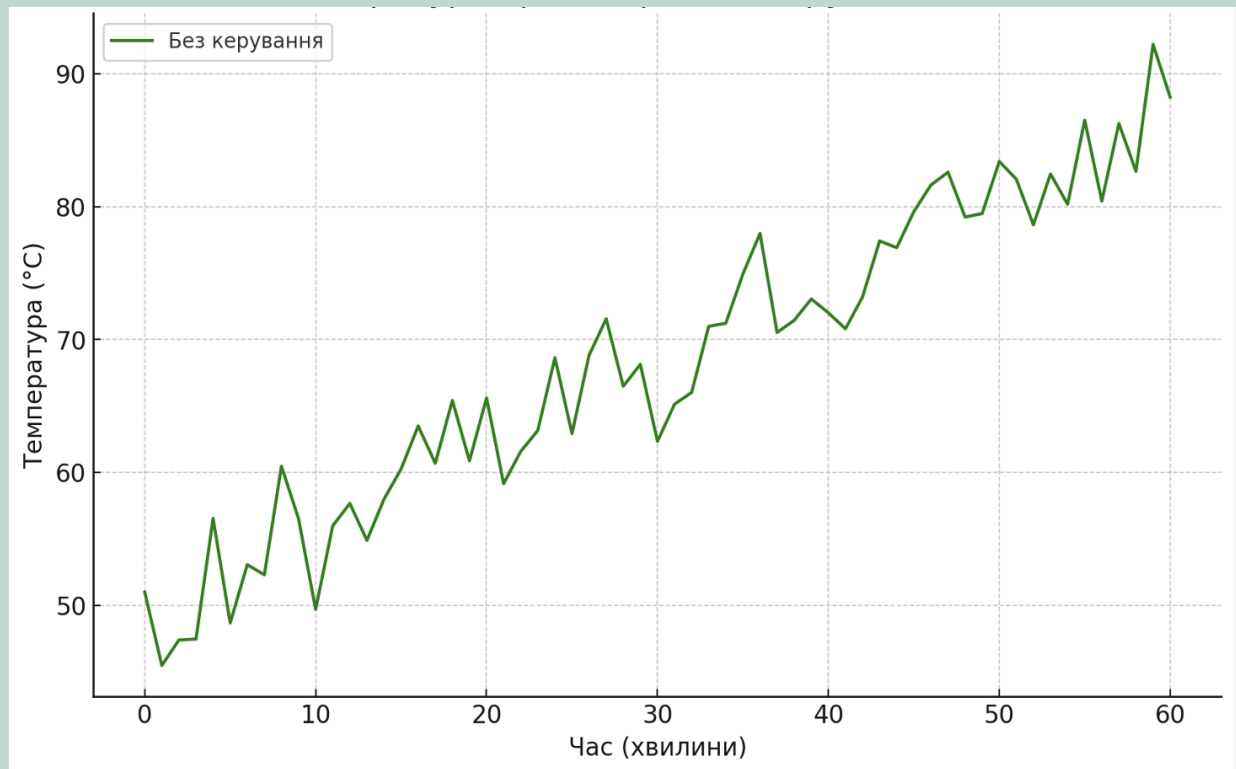
Пікове значення, перевищує 200 Вт

Середнє завантаження CPU без керування в часі



Пікове значення, перевищує 85-95%

Температура процесорів без керування в часі



30 хвилин експерименту температура перевищує 85–90°C

Енергоспоживання кожного вузла $P(t)$ в момент часу t моделюється як сума базового енергоспоживання та навантажувальних складових:

$$P(t) = P_{base} + \alpha \times CPU_{load}(t) + \beta \times Memory_{load}(t)$$

Значення параметрів приймалися наступними: $P_{base} = 80$ Вт, $\alpha = 0.8$, $\beta = 0.5$.

На основі показників навантаження процесора та пам'яті у часовому діапазоні 0–60 хвилин було побудовано відповідний ряд значень $P(t)$.

Розрахунок енергоспоживання до початку керування потужністю :

$$P_{до} = 80 + 0.8 \cdot 85 + 0.5 \cdot 110 = 210 \text{ Вт}$$

Після застосування методу передбачається зменшення навантаження CPU та пам'яті на 20%:

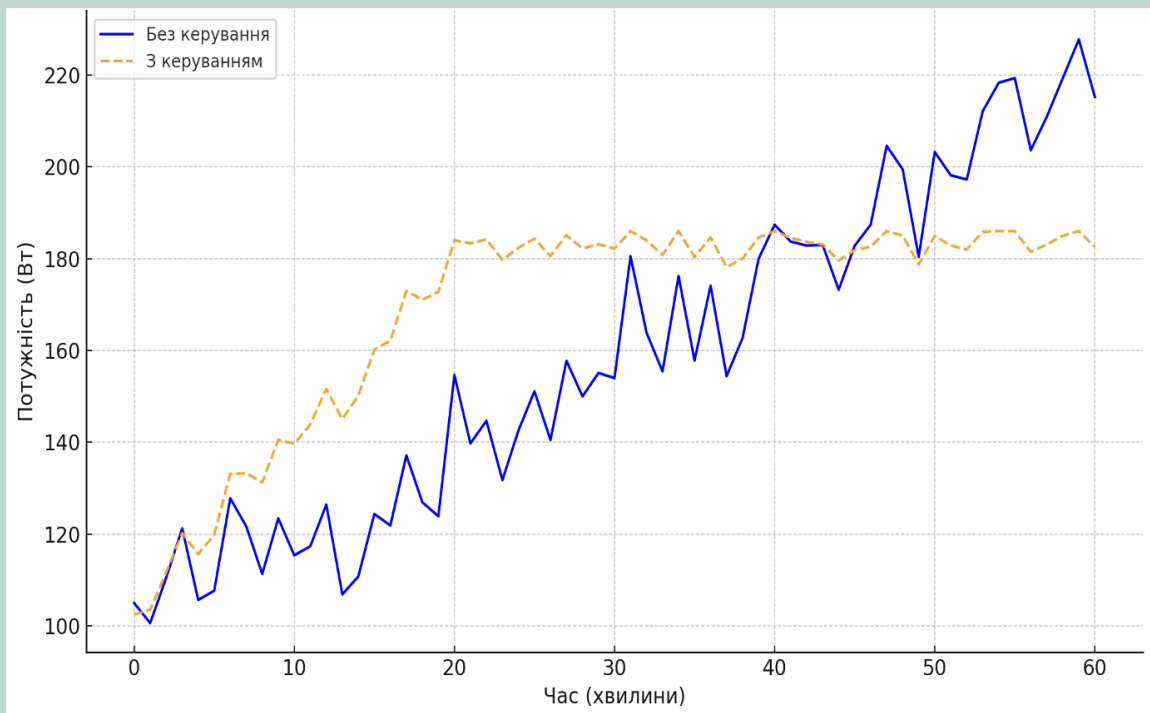
$$CPU_{new} = 85 \cdot (1 - 0.2) = 68\%$$

$$Memory_{new} = 110 \cdot (1 - 0.2) = 88\%$$

Підставляючи у формулу :

$$P_{після} = 80 + 0.8 \cdot 68 + 0.5 \cdot 88 = 184 \text{ Вт}$$

Порівняння графіків енергоспоживання до керування на після в часі



Енергоспоживання знижено з 220 Вт до 184 Вт

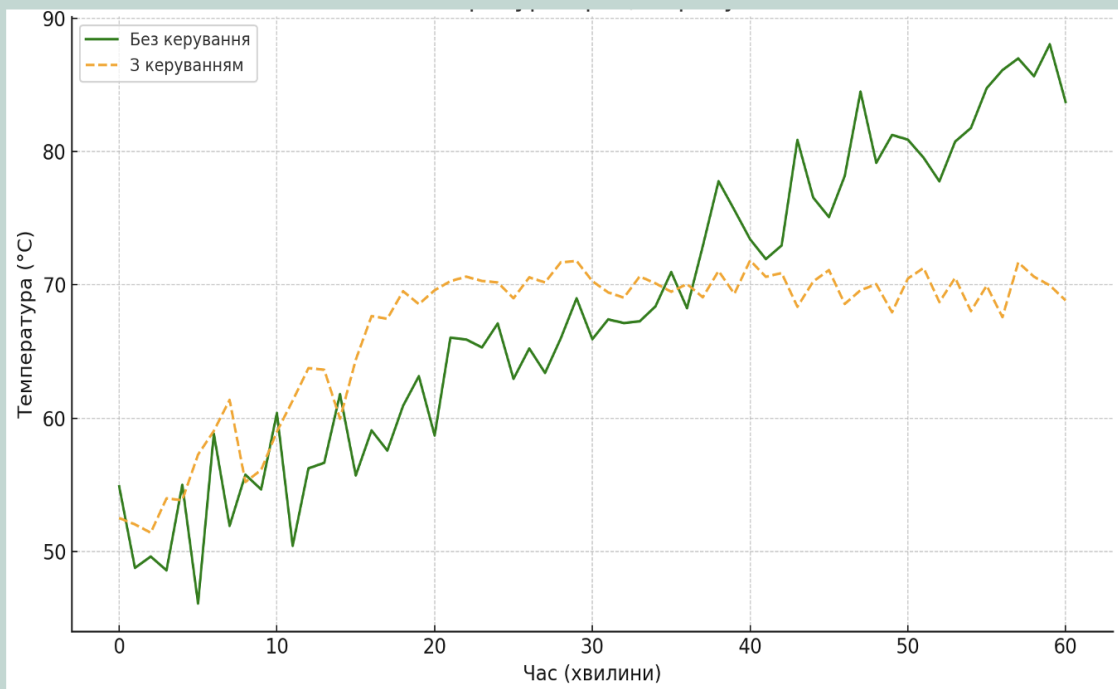
Температура апаратури оцінюється через лінійну залежність від CPU-навантаження:

$$T(t) = T_{start} + \frac{(T_{end} - T_{start})}{(CPU_{end} - CPU_{start})} \cdot (CPU_{new} - CPU_{start})$$

Підставляючи значення:

$$T_{new} = 50 + \frac{(82 - 50)}{(85 - 40)} \cdot (68 - 40) \approx 69.91^{\circ}C. \quad (3.14)$$

Порівняння графіків температури до керування на після



Температура процесора зменшилась з 82°C до 69.91°C

Для розрахунку зміни продуктивності використовується формула:

$$T_{task} = T_{base} \left(1 + \frac{CPU_{load}}{100} \right)$$

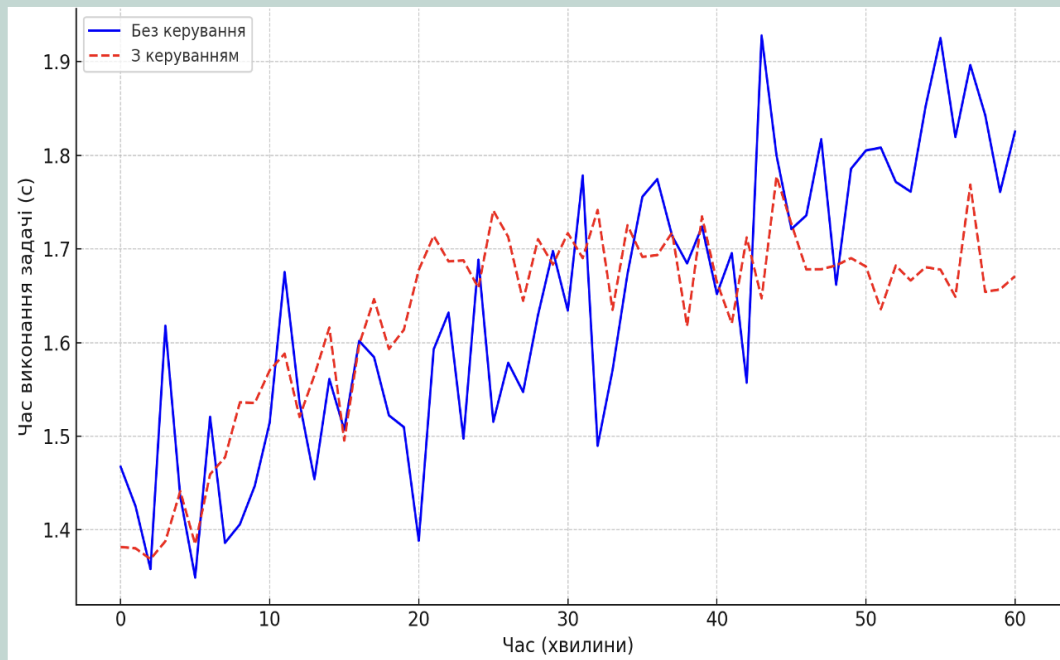
До керування:

$$T_{task,до} = 1.0 \cdot \left(1 + \frac{85}{100} \right) = 1.85с.$$

Після керування:

$$T_{task,після} = 1.0 \cdot \left(1 + \frac{68}{100} \right) = 1.68с.$$

Порівняння графіків середнього часу виконання задач до керування та після



Середній час виконання задач скоротився з 1.85 с до 1.68 с

КІНЦЕВІ РЕЗУЛЬТАТИ

У результаті реального застосування формул:

- Енергоспоживання знижено з 210 Вт до 184 Вт;
- Середній час виконання задач скоротився з 1.85 с до 1.68 с;
- Температура процесора зменшилась з 82°C до 69.91°C;
- Навантаження CPU знижено з 85% до 68%.

ВИСНОВКИ

У ході виконання дипломної роботи на тему «Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів» розроблено метод керування потужністю керування потужністю системи в комп'ютерній мережі та отримано такі результати.

1. Проаналізовано існуючі підходи до керування потужністю в мережевих системах.
2. Досліджено вплив основних профілів робочих параметрів на ефективність керування потужності.
3. Розроблено метод динамічного керування потужністю з урахуванням змінних параметрів мережі
4. Проведено моделювання та експериментальне оцінювання ефективності запропонованого методу.

ДЯКУЮ ЗА УВАГУ!

Готовий відповісти на ваші запитання



Дата звіту 5/21/2025
Дата редагування 5/21/2025

Документ прийнятий

Звіт подібності

метадані

Назва організації
Khmelnytskyi National University
Заголовок
Карвацький_Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів
Автор
Павло КАРВАЦЬКИЙ Науковий керівник / Експерт
підрозділ
Кафедра комп'ютерної інженерії та інформаційних систем

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



КП 1



КЦ

25

Довжина фрази для коефіцієнта подібності 2

15393

Кількість слів

122406

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв	Ⓡ	1
Інтервали	A→	0
Мікропробіли	␣	60
Білі знаки	Ⓡ	1
Парафрази (SmartMarks)	a	68

Anti-Plagiarism v-15.274 Educational

The maximum coincidence with one document 5.0%

Dictionaries check: en_US, ru_RU, ua_UA. Errors in the documents: 9%

ID: 241642 Title: МКР Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів Added in a DB: 2025-05-21 Authors: Павло КАРВАЦЬКИЙ Heads: Олег САВЕНКО Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	99321	810	7313 (7%)	74 (9%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Завідувачу кафедри КПС
доктору філософії, доценту
Ользі ПАВЛОВІЙ

Карвацького Павла Анатолійовича

ІІІІ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2М-23-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (StrikePlagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

21 травня 2025 року



Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Павло КАРВАЦЬКИЙ

Співавтор:

Назва: Карвацький_Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 6.8%

Коефіцієнт подібності 2: 0.9%

Мікропробіли: 60

Заміна букв: 1

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2025-05-21 20:35:57.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-05-21

Дата



Доцент Андрій Нічепорук

експерт

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів

Автор: Павло Карвацький

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Олег САВЕНКО, д.т.н., професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укріплення запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- окремі виявлені збіги є загальнонавчаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає менше 9% і адресується до джерел з інтернету та бібліотеки, що, з урахуванням наведених обґрунтувань, відповідає характеру завдання і свідчить на користь кваліфікаційної роботи.

Керівник роботи



Олег САВЕНКО

Гарант ОП



Олег САВЕНКО

Завідувач кафедри КІС



Ольга ПАВЛОВА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник: Павло КАРВАЦЬКИЙ

Тема: Метод керування потужністю системи в комп'ютерній мережі згідно профілів робочих параметрів»

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень -; кількість сторінок записки 73

1. Короткий зміст роботи та прийнятих рішень У роботі покращено метод керування потужністю в комп'ютерній мережі відповідно до динамічних профілів навантаження та робочих параметрів з метою підвищення енергоефективності та стабільності роботи системи.

2. Висновок про відповідність роботи дипломному завданню _____
Кваліфікаційна робота відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі подано об'єкт та предмет дослідження, мету, наукову новизну та практичну цінність роботи, а також характеристику структури роботи.

У першому розділі проведено аналіз відомих рішень щодо керування навантаження в мультикомп'ютерній системі.

У другому розділі здійснено визначення основ керування потужністю в комп'ютерних мережах за різними робочими параметрами.

У третьому розділі розроблено метод динамічного керування потужністю з урахуванням змінних параметрів мережі.

У четвертому розділі здійснено моделювання та експериментальне оцінювання ефективності запропонованого методу.

У висновках підведено підсумки досягнення результатів з розв'язання завдань дослідження.

4. Позитивні сторони роботи: _____

5. Негативні сторони роботи: немає.

6. Оцінка графічного оформлення та пояснювальної записки роботи: =

7. Відгук про роботу в цілому: Робота виконана на належному рівні.

8. Інші зауваження: =

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи вважаю, що робота заслуговує оцінки «добре» 3,75 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Бедратюк
Леонід Петрович, д.т.н., зав. кафедри ІПЗ ХНУ

“ 22 ” травня 2025р.

