

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління
(повна назва)

Кафедра _____ електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти _____ другий (магістерський)

_____ Методи розподілення віртуальних машин
_____ у хмарних системах

(тема)

здобувач _____ 2 _____ року навчання,
групи _____ СПМ-23-5
_____ Яків ЛОБАЧ
(власне ім'я, прізвище)

Спеціальність _____
_____ 123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми _____ освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма _____
_____ Системне програмування
(повна назва освітньої програми)

Керівник: _____ проф. Максим ВОЛК
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-наукова _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Лобачу Яківу Вадимовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Методи розподілення віртуальних машин у хмарних системах _____

затверджена наказом по університету від “ 21 ” _____ квітня _____ 2025 р. № _____ 296ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 16 червня 2025 р.

3. Вхідні дані до роботи _____

1. Методи розподілу ресурсів у хмарних системах _____

2. Провайдери хмарних сервісів _____

3. Методи віртуалізації _____

4. Технології управління хмарними системами _____

4. Перелік питань, що потрібно опрацювати в роботі _____

1 Аналіз предметної області _____

2 Методи розподілення віртуальних машин у хмарних системах _____

3 Експериментальні дослідження _____

4 Висновки _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____

Слайд-презентація – 12 слайдів _____

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

| Найменування розділу | Консультант (посада, прізвище, ім'я, по батькові) | Позначка консультанта про виконання розділу | |
|----------------------|--|---|------|
| | | підпис | дата |
| | | | |
| | | | |

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|---|--|--------------------------------|----------|
| 1 | Аналіз предметної області | 22.04.25-29.04.25 | |
| 2 | Розробка моделей | 30.04.25-05.05.25 | |
| 3 | Реалізація алгоритмів | 06.05.25-10.05.25 | |
| 4 | Розробка структури програмних засобів | 11.05.25-21.05.25 | |
| 5 | Розробка програмних модулів | 22.05.25-02.06.25 | |
| 6 | Оформлення матеріалів кваліфікаційної роботи | 03.06.25-05.06.25 | |
| 7 | Подання кваліфікаційної роботи керівникові та її попередній захист | 06.06.25-10.06.25 | |
| 8 | Подання кваліфікаційної роботи на рецензування | 11.06.25-12.06.25 | |
| | | | |
| | | | |

Дата видачі завдання 21 квітня 2025 р.

Студент _____

(підпис)

Керівник роботи _____

(підпис)

проф. Максим ВОЛК _____

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 57 с., 10 рис., 3 табл, 1 дод., 21 джерело.

ХМАРНІ ОБЧИСЛЕННЯ, РОЗПОДІЛ ВІРТУАЛЬНИХ МАШИН,
ОПТИМІЗАЦІЯ, ЕНЕРГОСПОЖИВАННЯ, БАЛАНСУВАННЯ
НАВАНТАЖЕННЯ, ЦЕНТРИ ОБРОБКИ ДАНИХ

У роботі досліджується проблема ефективного розподілу віртуальних машин у хмарних обчислювальних системах. Запропоновано метод оптимізації розміщення віртуальних машин, який зменшує енергоспоживання, підвищує ефективність використання ресурсів та покращує балансування навантаження в центрах обробки даних. Використання запропонованого підходу дозволяє мінімізувати кількість активних фізичних машин, скоротити час міграції віртуальних машин та забезпечити високу якість обслуговування. Проведені експериментальні дослідження підтвердили ефективність методики та її застосування в реальних умовах хмарних обчислень.

ABSTRACT

Master's thesis: 57 pages, 10 figures, 3 tables, 1 appendice, 21 sources.

CLOUD COMPUTING, VIRTUAL MACHINE DISTRIBUTION, OPTIMIZATION, ENERGY CONSUMPTION, LOAD BALANCING, DATA CENTERS

The paper investigates the problem of the efficient distribution of virtual machines in cloud computing systems. A method for optimizing the placement of virtual machines is proposed, which reduces energy consumption and resource efficiency and improves load balancing in data centres. The use of the proposed approach allows for minimizing the number of active physical machines, reducing the migration time of virtual machines, and ensuring high quality of service. Experimental studies have confirmed the effectiveness of the method and its application in real cloud computing conditions.

ЗМІСТ

| | |
|---|----|
| СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ..... | 8 |
| ВСТУП | 9 |
| 1 Аналіз предметної області..... | 11 |
| 1.1 Хмарні обчислення | 11 |
| 1.2 Алгоритми розміщення віртуальної машини..... | 12 |
| 1.3 Класифікація сучасних методів розподілення віртуальних машин..... | 15 |
| 1.4 Методи розподілення віртуальних машин | 17 |
| 1.5 Постановка мети та завдань дослідження | 21 |
| 2 Методи розподілення віртуальних машин у хмарних системах | 23 |
| 2.1 Загальна архітектурна модель системи розподілення віртуальних машин у хмарних системах..... | 23 |
| 2.2 Опис критеріїв ефективного розподілення віртуальних машин | 24 |
| 2.2.1 Максимізація продуктивності | 25 |
| 2.2.2 Мінімізація мережевого трафіку | 25 |
| 2.2.3 Максимізація якості обслуговування | 26 |
| 2.2.4 Мінімізація енергоспоживання | 26 |
| 2.2.5 Економічна оптимізація витрат | 27 |
| 2.2.6 Вибір критеріїв ефективності | 27 |
| 2.3 Метрики для оцінки ефективності розміщення віртуальних машин | 29 |
| 3 Експериментальні ДОСЛІДЖЕННЯ..... | 32 |
| 3.1 Методологія організації експериментів..... | 32 |
| 3.2 Опис експериментальної частини | 35 |
| ВИСНОВКИ..... | 46 |

| | |
|--------------------------------|----|
| | 7 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ | 48 |
| ДОДАТОК А..... | 51 |

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

ВМ – віртуальна машина
ОС – операційна система
ПЗ – програмне забезпечення
ЦОД – центр обробки даних
ЦП – Центральний процесор
API – Application Programming Interfase
DIS – Distributed Interactive Simulation
DEVS – Discrete Event System Specification
IaaS – Infrastructure as a Service
EKS – Elastic Kubernetes Service
ID – Identity
FCFS – First-Come First-Served
PaaS – Platform as a Service
PM – Physical Machine
ROI – Increase Return on Investment
SaaS – Simulation as a Service
SLA – Service-Level Agreement
SLLC – Shared Last Level Cache
VM – Virtual Machine
VMM – Virtual Machine Monitor
WAN – Wide Area Network (WAN)
QoS – Quality of Service

ВСТУП

Реаліями сьогодення є те, що дані оточують нас у всіх аспектах нашого життя. Для їх обробки потрібне високоякісне обладнання та значні обчислювальні потужності для підтримки якості доступу до ресурсів, що іноді дорогі і недоступні. Еволюція технологій призвела до розгортання хмарних обчислень для задоволення цих вимог. Вони пропонують такі переваги як доступність, масштабованість, надійність, відмовостійкість, мобільність, гнучкість та ефективне використання інфраструктурних ресурсів зі зниженою вартістю. Це дозволило їм зайняти міцне місце в повсякденному житті, обслуговуванні фірм та промисловості. Хмара пропонує широкі послуги клієнтам на основі їх потреб та служить основою для функціонування передових організацій. Один з найбільш використаних розрахунковий методів стягує з користувачів плату за обсяг спожитої послуги.

Хмарні обчислення [1] є причиною створення великої кількості центрів обробки даних, які складаються з тисяч серверів. Вони споживають величезну кількість електроенергії для охолодження, мережі тощо, що призводить до високих операційних витрат і викиду вуглекислого газу. Цю проблему вирішує ефективне покращення фізичної інфраструктури та ресурсів використання.

Інфраструктура як послуга є однією з важливих пропонованих послуг хмарних обчислень, яка дозволяє отримати віртуалізацію шляхом створення багатьох екземплярів віртуальної машини (VM, VM) на одній фізичній машині (PM) і допомагає вдосконалюватися методам використання ресурсів. Консолідація VM включає метод вибору більш відповідного алгоритму для міграції віртуальних машин і розміщення віртуальних машин на найбільш підходящому хості. Розміщення VM є частина міграції VM, що націлено на ефективне розміщення VM для покращити продуктивність, використання

ресурсів і зменшення споживання енергії в ЦОД без порушення SLA.

Щоб задовольнити зростаючі вимоги для обчислювальних ресурсів центри обробки даних містять велику кількість фізичних машин, що містять декілька віртуальних машин. Однак загалом, хмарні центри обробки даних не завжди можуть ефективно використовувати свої обчислюючи ресурси до їх загальної потужності. Було запропоновано кілька політик для покращення навичок використання енергії та обчислень у хмарних системах [2].

Розміщення віртуальних машин повинно обслуговуватися ефективним методом, який передбачає ефективне відображення віртуальних машин на фізичні машини. Проте доступність багатьох фізичних машин, що містять декілька віртуальних машин у центрі обробки даних зробило проблему розміщення віртуальної машини складною обчислювальною задачею. (NP повна задача). Для практичного вирішення знайшли широке застосування метаевристичні алгоритми, які вирішують складні проблеми із кількома та суперечливими цілями, наприклад, як проблема розміщення віртуальних машин. В кваліфікаційній роботі проведено таксономію метаевристичних алгоритмів для методів розміщення віртуальних машин та запропоновано свої рішення щодо архітектури та методів розподілення віртуальних машин у хмарних системах.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Хмарні обчислення

Хмарні обчислення пропонують послуги, які поділяють на три основні групи [3]: програмне забезпечення як послуга (SaaS), платформа як послуга (PaaS) та інфраструктура як послуга (IaaS), як на рисунку 1.1.

У SaaS програмне забезпечення, яке розгортається в Інтернет ліцензується постачальником програмного забезпечення клієнту або як послуга на вимогу, через модель "оплатити". Воно працює в хмарі та обслуговує різних користувачів. Наприклад, Gmail - популярний продукт SaaS. Доступ до додатків можна отримати через веб-браузер, мобільний телефон тощо. Немає спеціальних вимог до пристрою, браузера або мережі. Сервіси підтримують налаштування програми.

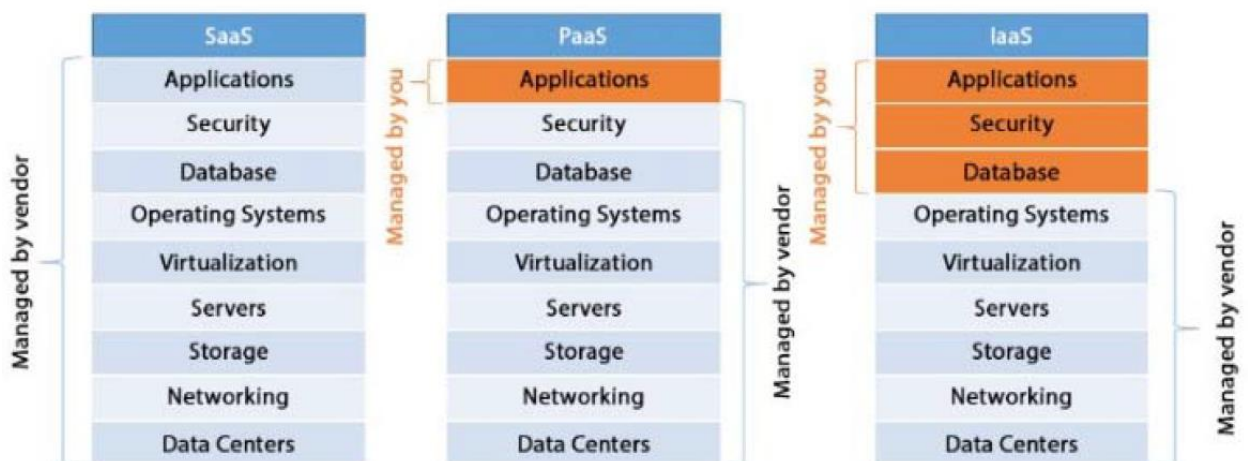


Рисунок 1.1 – Відмінності між шарами стеку, які використовуються різними моделями хмарних служб

PaaS побудований на вершині технології віртуалізації. Розробники цього рівня зосереджуються на створенні програми, не турбуючись про оновлення програмного забезпечення, балансування навантаження,

експлуатації системи, зберігання чи інші деталі про інфраструктуру. Це надає більш швидку, економічну модель для розробки та доставки послуги.

IaaS пропонує отримати віртуальний сервер і оплатити ресурс використання. Інфраструктура забезпечує обчислення як спільну послугу, що зменшує інвестиційні витрати, оперативне та технічне обслуговування обладнання. Він забезпечує такі ресурси, як сервери, мережеві компоненти і пристрої зберігання даних як послугу. Його можна отримати як приватну або як публічну інфраструктуру, або як комбінацію обох. IaaS дозволяє віртуалізувати апаратне забезпечення центру обробки даних і допомагає покращити використання ресурсів і окупність інвестицій.

1.2 Алгоритми розміщення віртуальної машини

Алгоритм розміщення віртуальної машини в центрі обробки даних виділяє різні ресурси, такі як пам'ять, пропускну здатність і обчислення тощо від фізичної машини до віртуальних машин, у таких спосіб мінімізує кількість фізичних машин та серверів. Міграція віртуальної машини складається з чотирьох етапів. На рисунку 1.2 показано кроки, які виконуються під час міграції.

На першому етапі обирається фізична машина, яка є недостатньо завантаженою або перевантаженою. Потім одну або кілька віртуальних машин обирається для міграції. Крок третій передбачає розміщення віртуальної машини, яка знаходиться на відповідній фізичній машині на іншій, яка задовольняє вимогам до ресурсів, таким як пам'ять, пропускну здатність, використання процесора та зовнішні сховища даних. Останнім кроком є міграція віртуальних машин на цільову фізичну машину.

У нашій роботі ми в основному зосереджуємось на третьому кроці, яким є вибір розміщення віртуальної машини.

Розміщення віртуальної машини є методом вибору найбільш підходящої фізичної машини для віртуальної машини. Якщо відображення

віртуальні машини фіксовані на всьому протязі обчислень, то це називається статичним розміщенням ВМ. Якщо місце розміщення дозволено змінювати відповідно до навантаження системи, то це називається динамічним розміщенням ВМ.

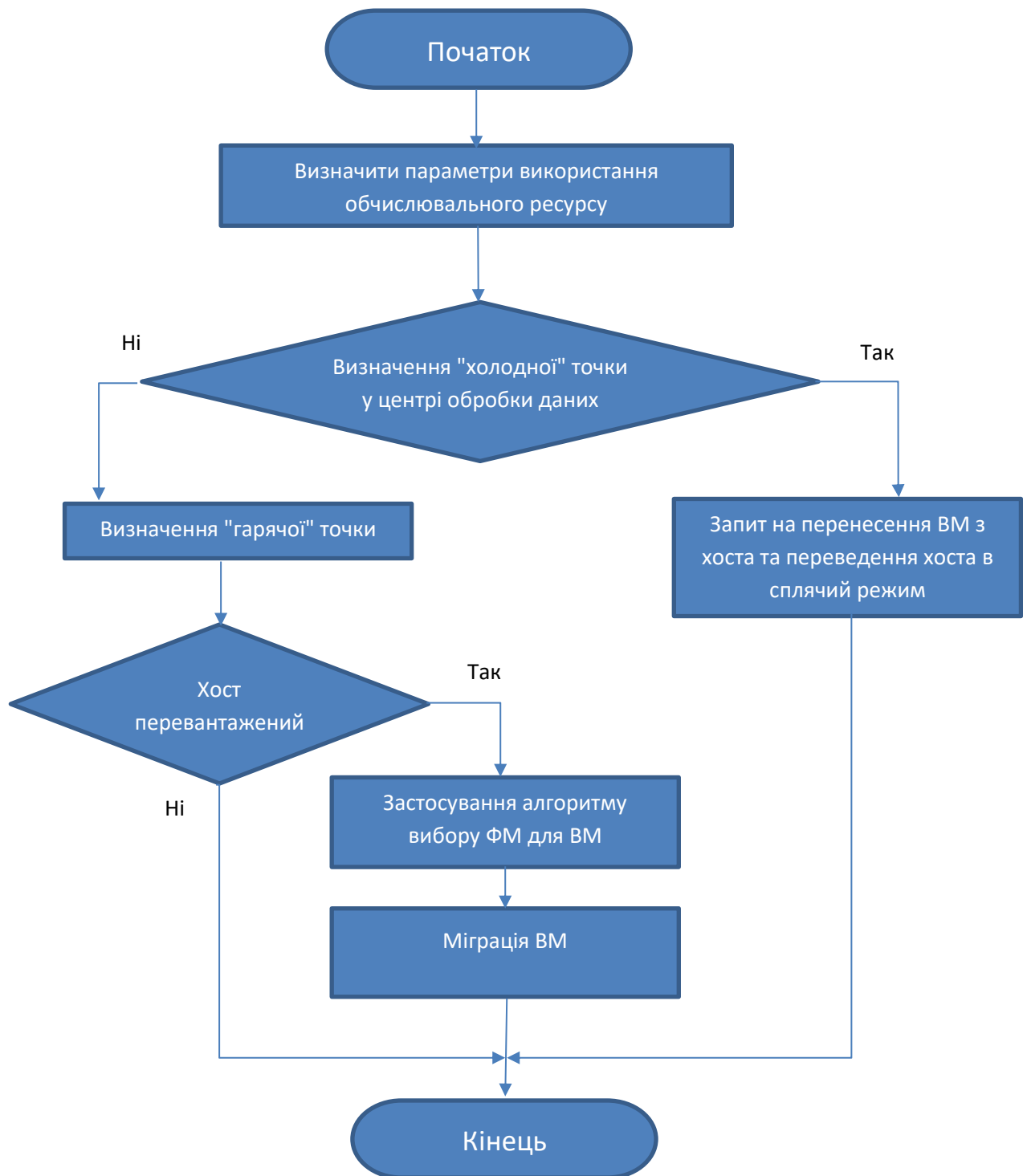


Рисунок 1.2 – Блок-схема процесу міграції віртуальної машини

Динамічне розміщення віртуальної машини класифікується далі у реактивне розміщення віртуальної машини та проактивне розміщення віртуальної машини та реактивне розміщення віртуальної машини лише після того, як система досягне небажаного стану, коли вносяться зміни до початкового розміщення.

У проактивному розміщенні віртуальної машини зміни початкового розміщення дозволено до того, як система досягне певного стану.

Наприклад, є 5 серверів, які не віртуалізовані і треба розмістити 5 програм, позначених App1 до App5, як показано на малюнку 1.3. Існують вимоги до ресурсів програми для кожного сервера. Необхідно перемістити програми з цих серверів на віртуалізовані сервери в хмарі, яка здатна виконувати кілька віртуальних машин паралельно. Ми можемо бачити на рисунку 1.3, що хости, які не є віртуалізовані використовуються не повністю.

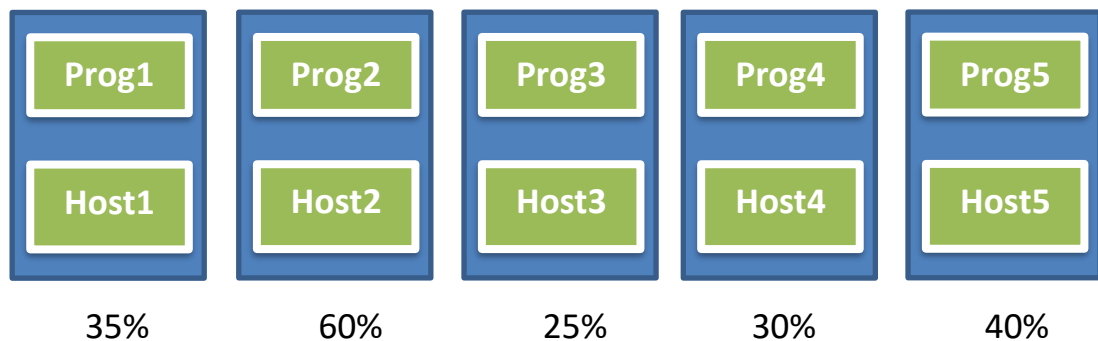


Рисунок 1.3 – Невіртуалізовані сервери

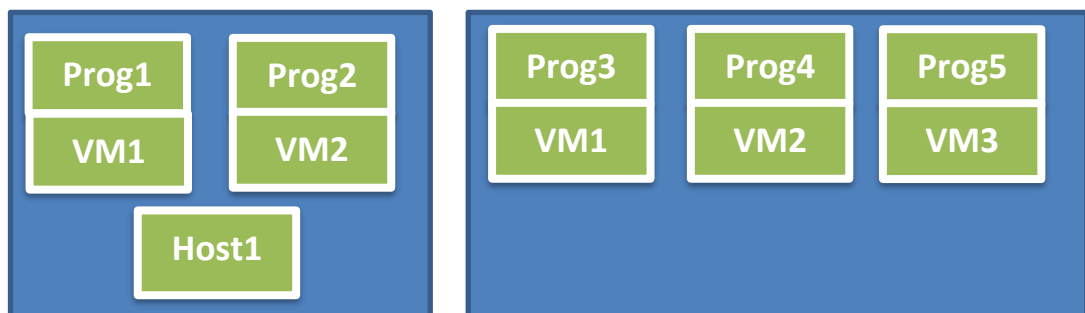


Рисунок 1.4 – Віртуалізовані сервери

Необхідну кількість хостів можна зменшити до 2 шляхом віртуалізації серверів, як показано на рисунку 1.4 і використовувати по максимуму ресурси цих двох хостів.

Таким чином консолідація віртуальних машин буде зменшувати споживання енергії та інші витрати. Це відображення VM на PM називається проблемою розміщення VM.

Ідеальний метод розміщення VM має наступні цілі, які іноді протирічать одна одній та зазначені нижче:

- покращення масштабованості центрів обробки даних;
- максимальне використання ресурсів;
- покращити балансування навантаження;
- зменшення споживання енергії;
- гарантована якість обслуговування (QoS) клієнтів;
- мінімізація трафіку центру обробки даних;
- запобігання перевантаженням у мережі центру обробки даних;
- висока продуктивність;
- забезпечення безпеки;
- збільшення рентабельності інвестицій (ROI);
- зменшення кількості активних елементів мережі;
- звести до мінімуму порушення SLA;
- зменшення кількості міграцій віртуальних машин у майбутньому.

1.3 Класифікація сучасних методів розподілення віртуальних машин

Залежно від мети розміщення, алгоритму розміщення можна класифікувати наступним чином (рисунок 1.5).

Методи, що мінімізують споживання електроенергії. Його метою є отримання відображення [2,4] таким чином, щоб сервери використовувалися максимально, роблячи систему енергоспоживання ефективною. Сервери, які не використовуються, вимикаються або переводяться у сплячий режим

залежно від умов навантаження для економії електроенергії [5].

Підхід, заснований на QoS програми. Він спрямований на відображення віртуальних машин на РМ, максимізуючи якість послуг (QoS), які надає хмарний постачальник. Його застосування призводить до зменшеної загальної вартості наданих послуг.

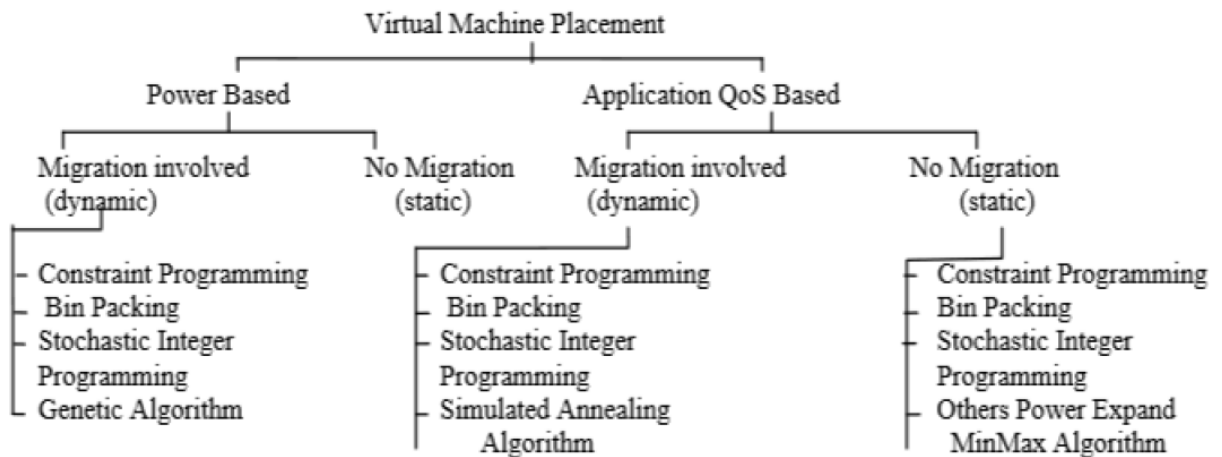


Рисунок 1.5 – Класифікація алгоритмів розміщення ВМ

Багато дослідників представили різні механізми розміщення віртуальних машин. Їх аналіз зосереджувався на різних точках зору проблеми розміщення ВМ. Так у [6] забезпечують комплексний огляд методів розміщення віртуальних машин з упором на споживання енергії. Автори в основному зосереджувалися на інтелектуальному управлінні для мінімізації енергії споживання ресурсів хмарних обчислень. Відповідно, вони переглянули класифікацію енергоефективного дизайну хмарних ресурси в різних операційних системах, апаратне забезпечення, центри обробки даних і віртуалізації. Вони виділили ряд вимог до ефективного рішення розміщення віртуальних машин, у тому числі під час зміни робочого навантаження, віртуалізації, забезпечення узгодження рівня обслуговування, забезпечення якість обслуговування різних програм на основі апаратних рішень.

У роботі [7] представлено комплексний огляд динамічної консолідації віртуальних машин. Вони зосередилися на кількісній класифікації. Спочатку

віртуальні машини розподіляються за допомогою механізмів консолідації в різні категорії, такі як статичні та динамічні, централізовані та розподілені архітектури. Зазначено, що методи, що засновані на основі порогових значень обмежені простором пошуку та знаходять оптимальні рішення за короткий проміжок часу. Вони також мають класифіковану консолідацію віртуальних машин на основі кількості міграцій віртуальної машини. Розвиток методи методи міграції VM і консолідація серверів отримано в [8]. Запропановано огляд методів живої міграції і таксономітриї середовища консолідації, архітектур системи розподілу, метод розподілу обчислювальних ресурсів, метод запуску міграції та їх моделі. Вони також запропонували розділити методи міграції віртуальних машин на основі на динамічної напруги, оптимізації смуги пропускання та масштабуванні, що дозволило оптимізувати енергоспоживання.

1.4 Методи розподілення віртуальних машин

В середовищі хмарних обчислень процес віртуалізації відіграє значну роль, розділяючи фізичні обчислювальні ресурси, такі як пам'ять, сховища даних та ЦП у кількох ізольованих середовищах виконання, які називаються віртуальними машини [9]. Ці віртуальні машини знаходяться в датацентрах, які географічно розподілені. Віртуальний машини з обчислювальними ресурсами, такими як пам'ять, ЦП, сховище та пропускна здатність мережі доступні та керовані ефективно користувачами відповідно до їхніх вимог та можливостям середовища хмарних обчислень. Розміщення віртуальної машини передбачає виділення відповідній віртуальній машині фізичної машини в хмарному центри обробки даних. Його також називають планом розміщення (схемою призначення) або зіставлення віртуальної машини з фізичною машиною.

Розглянемо найбільш відомі методи згідно класифікації (рисунок 1.5).

Метод упаковки контейнера (Bin Packing). Розміщення VM можна

сформулювати як проблема упаковки [10]. У проблемі упаковки контейнера мета полягає в тому, щоб звести до мінімуму розмір контейнеру для упаковки завдань різного об'єму. Пропонуються евристичні методи, оскільки ця задача відноситься до упаковки NP-повних. Віртуальні машини, які потрібно розмістити, вважаються об'єктами, які направлені на фізичні машини, які розміщуються у контейнерах. Вартість центру обробки даних знижується завдяки щільному розміщенню. Віртуальні машини розміщуються на найменшій кількості серверів. Методи спрямовані на скорочення робочих серверів, часу завершення та мінімізує кількість серверів для VM розміщення.

Прогноз попиту кожної VM обчислюється з урахуванням того, що на віртуальні машини сортується в порядку спадання обчислювальної складності. Потім береться кожна віртуальна машина зі списку та розміщується на першому PM, де він знаходиться. Цей алгоритм мінімізує кількість активних PM. Схеми розміщення VM в основному враховують енергетичні характеристики та параметри, пов'язані з ресурсами.

Генетичні алгоритми. Ця група алгоритмів використовує принципи розвитку природних генетичних популяцій. Початкова популяція (схема розподілу) складається з набору можливих рішень і використовується для формування наступного покоління (покращеної схеми розміщення). Розміщення VM можна сформулювати як генетичний алгоритм [11], в якому область рішення представлена як фізична машина з можливістю надання ресурсів. Функція придатності визначається кількістю контейнерів у рішенні. Мета полягає в тому, щоб мінімізувати кількість контейнерів.

Генетичний алгоритм групування також можна використовувати для віртуальної машини у проблемі розміщення, яка може враховувати додаткові обмеження. Цей алгоритм працює краще навіть на великих розмірах задачі розподілу ресурсів. Етапи генетичного алгоритму такі:

- початкова сукупність складається з набору рішень, що представлені хромосомами;

- вибирається найбільш підходяща селективна функція на основі вимог до схеми розподілу;
- проводиться оцінка кандидатів за допомогою функції селекції.
- обирається найкращий варіант;
- виконуються операції кросинговеру на парах хромосом для генерації наступної схеми розподілу; якщо кросовер виконується тільки в одній точці, це називається одноточковий кросовер.
- виконується операція мутації шляхом перестановки хромосом; це допомагає змінити значення генів хромосом і підтримувати генетичну різноманітність;
- вибираються найсильніші особи із нових та перевірка умов завершення розподілу та перехід до першого пункту, якщо умова завершення не виконана.

Ці методи допомагають розв'язати задачу багатоцільового розміщення та оптимізувати цілі, які конфліктують, наприклад скорочення споживання енергії, уникаючи недовантаження або перевантаження та можуть використовуватись для задач балансування навантаження [12].

Стохастичне програмування використовується для оптимізації задач, які включають невизначеність. Зазвичай в реальному світі є деякі невідомі параметри. Моделі стохастичного програмування приймають цей факт та використовують ймовірнісний підхід до даних, коли майбутні цілі або ресурси невідомі. Мета полягає в тому, щоб максимізувати очікування функції рішень з урахуванням випадкових змінних. Розміщення VM можна сформулювати як стохастичне ціле число, якщо відомі вимоги до ресурсів віртуальної машини. Необхідно знайти найбільш підходящу фізичну машину для зменшення втрати енергії та ресурсів.

Мурашиний алгоритм оптимізації був розроблений на основі поведінки справжніх мурах. Поки мурахі пересуваються в пошуках їжі, вони переміщуються з гнізда до джерела їжі, вони залишають феромони на своєму шляху. Феромон - це різновид хімічної речовини. Якщо нові або окремі

мурахі стикаються з феромоном, вони вирішують з високою ймовірністю по цій стежки. Завдяки власному феромону вплив посилюється. Якщо концентрація феромону на шляху висока, то ймовірність того, що мураха вибере цей шлях, висока. Якщо феромон є щільніший, він приваблює більше мурах. Якщо шлях короткий, випаровування феромону стає менше, а феромон стає щільнішим і інші мурахі, швидше за все, підуть цим шляхом через природу мурах. Якщо на проходження потрібно більше часу, посилені стежок стає менше. Конвергенції локального оптимального рішення уникають через процес випаровування феромонів. Мурахи подорожують усією мережею хмар, щоб розпізнати перевантажений або недостатньо завантажений хост у хмарній системі. Таблиця феромонів оновлюється, коли мурахі перетинаються і це допомагає контролювати використання ресурсів.

Алгоритм роботи наступний:

- ініціалізація мурахи феромонами;
- пошук мурах;
- вибір наступного стану;
- перевірка балансу навантаження;
- актуалізація таблиці феромонів;
- перевірка умови зупинки, якщо виконано, завершіть;
- виконання, інакше повторіть з кроку 2.

Програмування з обмеженнями [4] використовується для вирішення комбінаторної проблеми з пошуком. Розміщення VM можна сформулювати як розподіл з урахуванням деяких обмежень, що є перевагою методу. Цей метод використовується для вирішення задач з кількома цілями. Розміщення VM складається з двох етапів (локальні рішення та глобальні рішення). Обмеженнями у розміщенні VM може бути: обмеження ємності (для всіх характеристик певної фізичної машини, як-от ЦП, пам'ять і пропускна здатність, сума ресурсів, які використовуються усі віртуальні машини, що працюють на цій фізичній машині має бути меншим за загальну доступну

потужність цього фізичного машина); обмеження розміщення (усі віртуальні машини мають бути розміщені на хостах, які є доступними); обмеження SLA (VM слід розмістити на PM таким чином, щоб не було порушень SLA); обмеження якості послуг (QoS - під час розміщення VM певна якість обслуговування, такі як пропускна здатність, рівень помилок, доступність тощо враховуються). Програмування обмежень є корисним, оскільки вимоги VM відомі до обчислення функції вартості.

1.5 Постановка мети та завдань дослідження

У розробці схем розподілу VM є ряд проблем, які відображаються на методи планування. Деякі з них представлені нижче.

Не існує загальної моделі для представлення різних сценаріїв планування віртуальної машини. Додатки мають різні вимоги QoS на основі таких показників, як пропускна здатність, час відгуку та швидкість транзакції. Це залежить від типу програм і як вони використовуються. Моделювання та їх кількісна оцінка вимоги складні.

Параметризація моделі, тобто пошук відповідних значень для параметрів складний, коли розмір завдань великий. Для багатохмарних сценаріїв, які включають n хмарних провайдерів для m завдань, для отримання вираження потрібно призначити параметри віртуальних машин, та розуміти накладні витрати на міграцію віртуальної машини, ігноруючи можливі зміни самої віртуальної машини. Існує потреба в механізмах, які допомагають автоматично фіксувати ці значення.

Конфліктні цілі виникають при наявності кілька цілей, які можуть бути суперечливим.

Безперервна оптимізація виникає через динамічний характер хмари, розподіл ресурсів потрібно регулярно оновлювати за причин продуктивності, невдач та збоїв тощо. Важко прийняти рішення коли і як переналаштувати хмару для адаптації до динамічних змін.

Метою роботи є підвищення ефективності управління ресурсами в хмарних обчислювальних системах шляхом оптимізації процесу розміщення віртуальних машин, що дозволяє зменшити енергоспоживання, скоротити час міграції та покращити рівень обслуговування користувачів.

Для реалізації мети роботи вирішені наступні задачі:

- проведено аналіз існуючих методів розподілу віртуальних машин у хмарних середовищах;
- розроблено алгоритму оптимізованого розміщення віртуальних машин;
- проведена оцінка ефективності запропонованого підходу на основі експериментальних досліджень;
- виконано порівняння отриманих результатів з існуючими методами.

Об'єктом дослідження виступають процеси управління ресурсами у хмарних обчислювальних системах.

Предмет дослідження – методи оптимізації розподілу віртуальних машин у хмарних обчислювальних середовищах.

2 МЕТОДИ РОЗПОДІЛЕННЯ ВІРТУАЛЬНИХ МАШИН У ХМАРНИХ СИСТЕМАХ

2.1 Загальна архітектурна модель системи розподілення віртуальних машин у хмарних системах

Аспекти запропонованого методу прогнозування енергоспоживання ВМ реалізовані у проактивний спосіб. На рисунку 2.1 показано загальну структуру для запропонованої системи, яка зосереджена на прогнозуванні використання потужності віртуальної машини. Запропонована структура складається з різних компонентів, які включають модуль хмарної інформаційної служби, модуль надання ресурсів, модуль машинного навчання та модуль прийняття рішень.

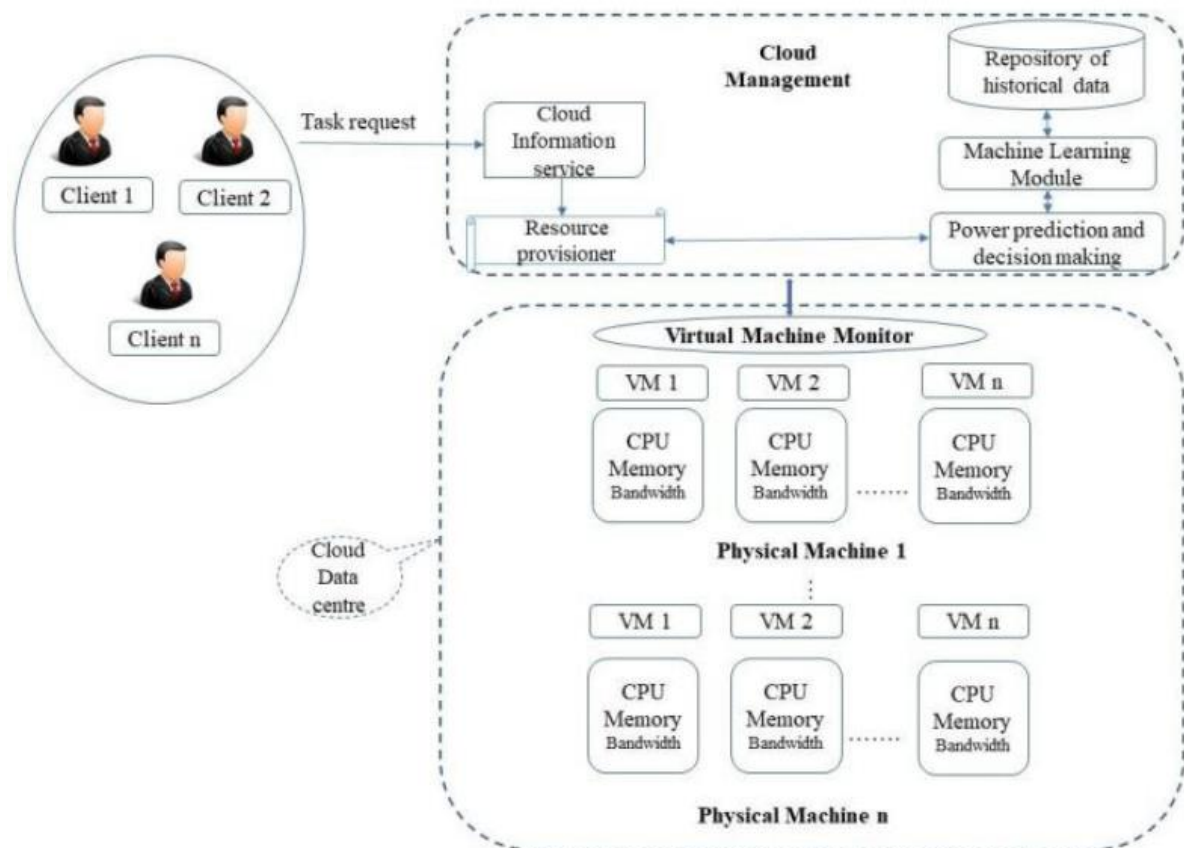


Рисунок 2.1 – Структура запропонованої системи

Кілька запитів VM від клієнта реєструються в модулі хмарної інформаційної служби для розгортання їх системи та програми. Модуль надання ресурсів розподіляє ресурси для віртуальних машин на основі рішення хмарного менеджера, коли це необхідно. Цей модуль відповідає за задоволення запиту на обслуговування для клієнтів відповідно до угоди про рівень обслуговування (SLA).

Модуль ML перевіряє сховище історичних даних віртуальних машин, а потім вибирає дані для фази навчання та тестування. Вони витягуються модулем прийняття рішень для прогнозування потужності. Адміністрація хмари контролює інші модулі та приймає рішення у відповідній ситуації.

Хмарний центр обробки даних складається з підключених хостів, у яких кожному хосту призначено кілька віртуальних машин. Монітор віртуальної машини (VMM) — це рівень, який контролює кожну віртуальну машину, розташовану на фізичних машинах. VMM отримує результат від хмарного менеджера та призначає віртуальну машину бажаному РМ.

Несподіване створення екземпляра віртуальної машини на фізичному хості або призначення завдання існуючій віртуальній машині призводить до зміни атрибутів віртуальної машини; наступні коливання споживання електроенергії відбуваються у відповідному фізичному хості. У цьому сценарії аномалії потужності можна регулювати за допомогою передбачення в будь-який момент на основі історичних даних до зміни споживання електроенергії.

2.2 Опис критеріїв ефективного розподілення віртуальних машин

Хмарні центри обробки даних зберігають кілька обчислювальних ресурсів, таких як пам'ять, сховище даних, процесори, графічні процесори і мережна пропускна здатність. У середовищі хмарних обчислень складно ефективно використовувати всі обчислювальні ресурси. Розглянемо різні цільові функції максимізації використання ресурсів, які зазначені нижче:

- максимізація використання ресурсів;
- мінімізація втрат ресурсів;
- мінімізація максимального середнього використання;
- максимізація еластичності.

2.2.1 Максимізація продуктивності

Максимізація продуктивності також розглядається як важливий критерій для визначення політики розміщення віртуальної машини. Значущі цільові функції для продуктивності максимізації перераховані нижче:

- мінімізація загального часу виконання роботи;
- мінімізація суперечок спільного кешу останнього рівня (SLLC);
- максимізація показників безпеки;
- мінімізація перешкод ресурсу;
- максимізація QoS;
- мінімізація часу плану розгортання;
- максимізація задоволення вимог ЦП;
- максимізація доступності.

2.2.2 Мінімізація мережевого трафіку

Багато дослідників також зосереджувалися на мінімізації мережного трафіку для вирішення проблем взаємодії віртуальних машини на різних вузлах. Найпоширеніші цільові функції, які розглядаються для оптимізації у цьому напрямку перераховані нижче:

- мінімізація коефіцієнта навантаження для найгіршого випадку;
- мінімізація WAN зв'язку;
- мінімізація загальних витрат на зв'язок;
- мінімізація вартості вузла;
- мінімізація витрат мережі;

- максимізація продуктивності мережі;
- мінімізація використання мережі;
- мінімізація кількості міграцій;
- мінімізація накладних витрат на міграцію;
- мінімізація часу міграції;
- мінімізація перевантаження каналів;
- мінімізація наскрізної затримки;
- мінімізація доступу до даних;
- мінімізація часу передачі даних;
- максимізація Cloud QoE (мінімізація часу відгуку);
- мінімізація часу відгуку хмарного сервісу;
- мінімізація середньої затримки трафіку.

2.2.3 Максимізація якості обслуговування

Багато дослідників вирішували питання розміщення віртуальної машини проблему шляхом розгляду різних обмежень [9]. Одним з суттєвих обмежень є забезпечення якості послуг хмарних клієнтів. Найпоширеніші цільові функції максимізація якості обслуговування включає в себе наступні [13]:

- зменшення ресурсних перешкод;
- висока доступність, підвищення продуктивності;
- мінімізація перешкод ресурсу;
- надійність.

2.2.4 Мінімізація енергоспоживання

Більшість досліджень у цій галузі зосереджено на мінімізації споживання енергії для розподілу віртуальних віртуальних машин, використовуючи методи розміщення на різних рівнях [2,14,15].

Найпоширеніші цільові функції, пов'язані з мінімізацією енергії описані нижче:

- енергоспоживання рівня WDM;
- мінімізація енергоспоживання;
- мінімізація кількості ПМ;
- мінімізація енергоспоживання мережі;
- мінімізація енергоспоживання рівня IP;
- мінімізація енергоспоживання;
- максимізація енергоефективності;
- мінімізація енергоспоживання центру обробки даних.

2.2.5 Економічна оптимізація витрат

Мало дослідників розглядали економічну вартість визначення політики розміщення віртуальної машини. Вони в основному зосереджені на мінімізації різних витрат і збільшенні повернення інвестицій для центрів обробки даних. Найзначніші цільові функції, які дослідники оптимізують:

- мінімізація загальних витрат на інфраструктуру;
- мінімізація витрат на розсіювання тепла;
- мінімізація порушень SLA;
- мінімізація вартості сервера;
- мінімізація вартості бронювання;
- мінімізація операційних витрат;
- мінімізація витрат на електроенергію;
- максимізація економічного доходу.

2.2.6 Вибір критеріїв ефективності

Є кілька параметрів і міркувань у прийнятті рішень про те, куди і коли розподілити ресурси в хмарному середовищі. Основні аргументи є такими.

Продуктивність. Центри обробки даних дедалі більше працюють для віртуалізації та консолідації щодо підтримки великої кількості програм, які працюють одночасно. Результати суттєво відрізняються залежно від різної стратегії планування віртуальних машин. Якщо задіяно кілька хмарних провайдерів, продуктивність викликає додаткове занепокоєння.

Енергоефективність. Зі збільшенням інтересу до екоефективності технологій, загальна ефективність з боку використання, потужності і вартісті стали головною проблемою. Це зазвичай суперечить продуктивності хмарних обчислень.

Витрати. У моделі цін домінували фіксовані ціни на ранній стадії, але зараз тенденції показують зростання інтересу до динамічного ціноутворення. Внутрішні неявні витрати на планування розподілу віртуальних машин, наприклад, перешкоди та накладні витрати віртуальної машини, синхронізація та врахування віртуальних машинах, які працюють на тієї самої фізичної машині, також повинні бути прийнято до уваги.

Розташування. Віртуальні машини мають бути розташовані поблизу користувачів з міркувань зручності використання та доступності. Проте внаслідок наприклад, юридичних проблем та міркувань безпеки, географічне розташування може стати обмеженням для оптимального планування. Це стосується обох сторін: постачальників послуг, які використовують ресурси з кількох постачальників хмар і хмарних постачальників з географічно розподіленими центрами обробки даних.

Надійність і постійна доступність. Сервіси забезпечення надійності і доступності є частиною головних цілей планування розміщення віртуальних машин. Віртуальні машини реплікуються в кількох місцях для досягнення цього. Слід враховувати такі фактори, як важливість служби, інкапсульованої у віртуальних машинах, частота використання та надійність центрів обробки даних.

2.3 Метрики для оцінки ефективності розміщення віртуальних машин

Основною метрикою для оцінювання будемо вважати енергоспоживання. Він полягає в оцінці енергоспоживання [16] з центру обробки даних. Пропорційно із завантаженням ЦП зростає споживання електроенергії [17]:

$$P(C) = (P_{busy} - P_{idle}) \cdot C + P_{idle}, \quad (2.1)$$

де $P(C)$ – це розрахункове споживання електроенергії для використання C центральних процесорів;

P_{idle} – електроенергія, споживана під час, коли сервер був неактивним;

P_{busy} – споживання енергії під час розміщення та виконання ВМ.

Вторим критерієм введемо загальний час міграції. Значний обсяг даних [16] копіюється з пам'яті однієї фізичної машини на іншу під час виконання процесу живої міграції:

$$T_{mig} = V_{mig} / band, \quad (2.2)$$

де T_{mig} – це загальний час міграції, який залежить від обсягу пам'яті для передачі від хоста до цілі;

$band$ – та пропускна здатність мережі.

Для хорошого алгоритму розміщення ВМ загальний час міграції має бути меншим.

Час простою (downtime). Коли віртуальна машина запускається на новому хості під час живої міграції, є затримка під час синхронізації з іншими хостами та системою управління розподіленими обчисленнями. Це призводить до появи часу простою, який визначається наступним чином:

$$T_{idle} = (d \cdot l \cdot t_n) / band, \quad (2.3)$$

де T_{idle} – це час простою;

d – кількість сторінок в контейнері віртуальної машини;

l – розмір однієї сторінки;

t_n – час передачі одного фрейму згідно мережному протоколу;

$band$ – та пропускна здатність мережі.

Для ефективного алгоритму розміщення віртуальних машин, час простою має бути менше.

Останнім критерієм розглянемо рівень порушення SLA. Угода про рівень обслуговування (SLA) — це рівень обслуговування, який очікується від постачальника послуг. Порушення SLA має бути враховано під час розміщення VM для надання гарантованих послуг. Порушення SLA (SLAV) визначемо як

$$S = S_{mig} \times S_h, \quad (2.4)$$

де S_{mig} – це зниження продуктивності через міграції;

S_h – це порушенням SLA на активному хості.

У свою чергу, зниження продуктивності через міграцію визначається наступним виразом:

$$S_{mig} = \frac{\sum \frac{C_{dj}}{C_{rj}}}{M}, \quad (2.5)$$

де M – загальна кількість віртуальних маши;

C_{dj} – продуктивність деградації j -ої VM за рахунок міграцій;

C_{rj} – сумарна ємність ЦП, яку використала запитуваний j -а VM протягом свого життя.

Порушення SLA на активному хості:

$$S_h = \frac{\sum \frac{T_{si}}{T_{ai}}}{N}, \quad (2.6)$$

де N – кількість хостів;

T_{si} – загальний час, коли на хості спостерігалось 100% використання центрального процесора, що призвело до порушення SLA;

T_{ai} – загальний час, протягом якого хост був активним для обслуговування віртуальної машини.

Загальну продуктивність системи можна покращити, а трафік віртуальних машин можна зменшити шляхом ефективної схеми розподілу ресурсів. Це, у свою чергу, може зменшити кількість міграцій віртуальних машин у майбутньому.

Кожен з алгоритмів розміщення VM зможе працювати краще за певних умов. Методи розподілу повинні бути обрані таким чином, щоб вони приносили користь постачальнику хмари та користувачу без порушення SLA.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Методологія організації експериментів

У великомасштабних хмарних середовищах з великою кількістю високопродуктивних обчислювальних пристроїв неефективність використання енергії стає серйозною проблемою через недостатнє використання ресурсів. Віртуалізація вирішує цю проблему, дозволяючи розміщувати декілька екземплярів віртуальних машин на одному фізичному сервері, тим самим зменшуючи операційні витрати, енергоспоживання та втрату ресурсів. Для подальшого покращення використання ресурсів недостатньо використовувані сервери можна перевести в стан низького енергоспоживання.

VMР має на меті мінімізувати кількість фізичних серверів, необхідних для розміщення віртуальних машин, під час створення карт міграції для живої міграції. Однак надмірна міграція віртуальних машин під час консолідації може погіршити продуктивність програми та час відповіді. Крім того, спроби мінімізувати споживання енергії за допомогою консолідації віртуальних машин можуть ненавмисно порушити SLA між постачальниками послуг і клієнтами.

На рисунку 3.1 наведено приклад покращеного розміщення віртуальної машини з метою зменшення енергоспоживання та втрати ресурсів. На цьому рисунку дев'ять віртуальних машин розміщено на чотирьох різних фізичних машинах, які певною мірою використовують ресурси. Видно, що ресурси не використовуються належним чином.

Одним із способів покращити використання ресурсів є міграція віртуальних машин із сервера 1 на сервер 2 і з сервера 4 на сервер 3, щоб сервери 2 і сервери 3 стали більш завантаженими та добре збалансованими щодо використання ресурсів, тоді як сервери 1 і 4 були вимкнені для економії

енергії. Припускається, що всі сервери на рисунку належать до Intel Xeon 8171M 2,1 ГГц (Skylake) [x86-64] або Intel Xeon E5-2673 версії 3 2,4 ГГц (Haswell) [x86-64], де енергоспоживання на різних рівнях використання вказано в таблиці 3.1.

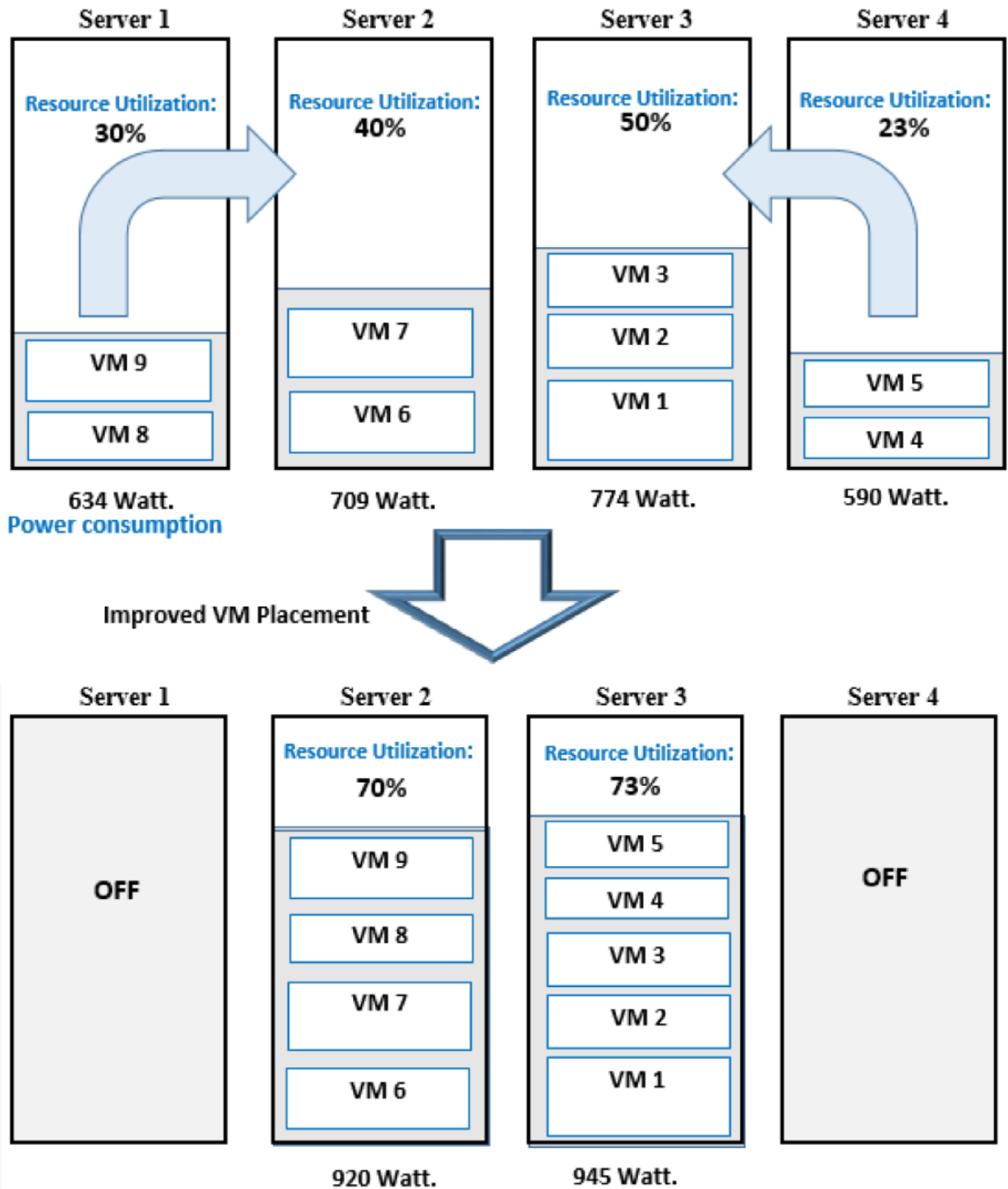


Рисунок 3.1 – Приклад покращеного розміщення віртуальних машин на фізичних машинах

Таблиця 3.1 – Споживання електроенергії у ватах різними серверами при різних рівнях використання.

| Процент використання процесора (%) | Фізична машина 1, Вт | Фізична машина 2, Вт |
|------------------------------------|----------------------|----------------------|
| 0 | 380 | 200 |
| 10 | 585 | 312 |
| 20 | 683 | 445 |
| 30 | 765 | 560 |
| 40 | 856 | 645 |
| 50 | 959 | 786 |
| 60 | 1210 | 902 |
| 70 | 1450 | 1035 |
| 80 | 1978 | 1135 |
| 90 | 2135 | 1221 |
| 100 | 2298 | 1350 |

Включення коефіцієнта втрати ресурсів у розрахунок енергоефективності допомагає зменшити потенційну міграцію віртуальних машин, спричинену дефіцитом ресурсів на серверах хостингу. Крім того, це сприяє ефективному використанню ресурсів РМ і підтримує баланс між ресурсами, що залишилися, у багатьох вимірах.

Коли дві віртуальні машини працюють на одному сервері, використання ЦП і пам'яті сервера приблизно оцінюється шляхом додавання окремих використання ЦП і пам'яті віртуальних машин. Наприклад, якщо одна віртуальна машина запитує 30% ЦП і 40% пам'яті, а інша віртуальна машина запитує 20% ЦП і 20% пам'яті, сумарне використання сервера, на якому розміщені обидві віртуальні машини, буде оцінено як 50% ЦП і 60% пам'яті відповідно, відображаючи підсумовування векторів використання.

Щоб запобігти повному використанню ЦП і пам'яті сервера (100%), необхідно встановити верхню межу використання ресурсів на сервер, яка позначається пороговим значенням. Цей запобіжний захід базується на розумінні того, що робота на повну потужність може призвести до значного погіршення продуктивності, а також тому, що динамічна міграція віртуальної машини споживає деякі ресурси процесора на вузлі міграції.

3.2 Опис експериментальної частини

Розглянемо алгоритм 1, який ґрунтується на попередньо розглянутих у розділі 2 параметрах та рівняннях. Загалом, алгоритм, як і будь-який типовий VMP, приймає список РМ і список VM як вхідні дані, а потім створює набір пар VM–РМ як вихідні дані, у якому кожна VM розміщується на певних хостах відповідно до обраної цільової функції.

По-перше, віртуальні машини сортуються в порядку зменшення відповідно до вимог ЦП. Для кожної віртуальної машини в списку віртуальної машини алгоритм перевіряє набір активних серверів, які мають достатньо ресурсів для задоволення запиту віртуальної машини, і обчислює енергоефективність для кожної з них. Для розрахунку енергії споживання для кожного хоста алгоритм визначає зміну енергоспоживання до та після розміщення VM. Таким чином, хост з найменшим збільшенням енергоспоживання матиме кращі шанси на хостинг.

Такий крок допомагає підібрати сервер з найвищою енергоефективністю. Це означає, що господар із вищим значенням продуктивності є кращим, оскільки він може розмістити більше віртуальних машин із меншим збільшенням енергоспоживання порівняно з іншими РМ. Це призводить до мінімізації загального енергоспоживання шляхом використання якомога меншої кількості хостів. Після цього обчислюється потенційний вплив порушення SLA до та після розміщення віртуальної машини.

Коефіцієнт використання ресурсів номінованих серверів обчислюється з урахуванням двох вимірів, центрального процесора та оперативної пам'яті. Цей параметр важливий для вибору відповідного РМ з точки зору використання ресурсів у багатьох вимірах. Крім того, це допомагає збалансувати втрату ресурсів РМ, віддаючи пріоритет тим, які використовують більше ресурсів.

Зазначений алгоритм представлено на рисунку 3.2.

Algorithm 1 VMP-ER Placement Algorithm.

Require: \mathbb{H} and \mathbb{V}

Ensure: \mathbb{Z} : where \mathbb{Z} is $\{(v_1, h_1), (v_2, h_2), (v_3, h_3), \dots\}$

1: **for all** $v_i \in \mathbb{V}$ such that $\mathbb{M}(v_i) \geq \mathbb{M}(v_{i+1})$ **do**

2:

$$h_{active}^{ch} = \begin{cases} \arg \max_{h_j \in \mathbb{A}(\mathbb{H})} \mathbb{E}^e(h_j), & \text{if } \mathbb{R}^q(v_i) \leq \mathbb{R}^f(h_j) \\ -\infty, & \text{otherwise} \end{cases}$$

3: **if** $h_{active}^{ch} \neq -\infty$ **then**

4: $\mathbb{Z} \cup (v_i, h_{active}^{ch})$

5: **else**

6: $h_{in-active}^{ch} = \arg \max_{h_k \notin \mathbb{A}(\mathbb{H})} \mathbb{E}^e(h_k)$

7: $\mathbb{Z} \cup (v_i, h_{in-active}^{ch})$

8: **end if**

9: **end for**

10: **return** \mathbb{Z}

Рисунок 3.2 – Алгоритм розподілу віртуальних машин

Показник енергоефективності для кожного сервера буде отримано на основі енергоефективності, використання ресурсів і можливого виникнення SLA, які сприяють вибору РМ. Для розміщення віртуальної машини буде вибрано найбільш підходящу, поки оновлюється з'єднання між віртуальною

машиною та РМ (рядок 4). Якщо немає активного сервера для розміщення віртуальної машини, алгоритм вибирає відповідний хост із набору неактивних серверів, який має найбільш енергоефективний показник (рядок 6). Вибраний хост буде пов'язано з віртуальною машиною (рядок 7). Нарешті, коли всі віртуальні машини розподіляються за їхніми хостами, алгоритм повертає остаточний набір пар віртуальної машини та фізичної машини.

Нехай у наборі n віртуальних машин V і m кількість РМ у наборі H . Алгоритм приймає $\mathcal{O}(n \log n)$, щоб відсортувати віртуальні машини в порядку спадання на основі вимог до ресурсів. Для кожної віртуальної машини алгоритм перевіряє всі активні РМ, щоб знайти відповідний хост, а потім перевіряє всі неактивні, якщо немає доступного активного хоста. Усі розрахунки для $E(h)$ займають постійну кількість часу ($\mathcal{O}(1)$) і його можна ігнорувати. Таким чином, загальний час для сортування віртуальних машин і перевірки всіх хостів займає $\mathcal{O}(n \log n + nm)$.

Оскільки кількість РМ менша за кількість віртуальних машин у хмарній системі, ми можемо виразити складність алгоритму як $\mathcal{O}(n \log n + n^2)$. Зі збільшенням n загальну складність можна спростити до $\mathcal{O}(n^2)$

Основна концепція включення $Rw(h)$ є фактором у виборі РМ, що покращує використання ресурсів фізичної машини в різних вимірах, одночасно забезпечуючи збалансований розподіл ресурсів, які залишилися. Це також допомагає зменшити непотрібні потенційні міграції через суперечку за ресурси, таким чином підвищуючи оптимізацію енергоспоживання для всього центру обробки даних.

$R(h)$ являє собою нормовану втрату ресурсів хоста, $(1 - Rw(h))$ відображає нормоване використання ресурсів, яке ми включаємо в наш розрахунок коефіцієнта енергоефективності. Це означає, що хост, який має прийнятний рівень використання ресурсів, матиме пріоритет для розміщення віртуальної машини, що призведе до більшого використання хостів, мінімізуючи загальну кількість увімкнених серверів. Це вказує на зменшення

загального енергоспоживання центру обробки даних через зменшення кількості активних хостів.

Щоб зменшити енергоспоживання центру обробки даних, перевантажені та недовантажені машини не розглядаються для розміщення. Це виключення відбувається, коли залишкова ємність фізичної машини перевищує певний поріг, перевищуючи верхній поріг або опускаючись нижче нижнього. Отже, віртуальні машини на недостатньо завантажених РМ буде перенесено на інші хости, що дозволить вимкнути ці машини для економії енергії.

3.3 Опис результатів проведення експериментів

Сучасний центр обробки даних характеризується своєю гетерогенністю, що складається з різних поколінь РМ з різними конфігураціями, зокрема за швидкістю та потужністю процесора. Ці сервери зазвичай інтегруються в центр обробки даних поступово, часто замінюючи старі застарілі машини в інфраструктурі [18]. Таким чином, ми розглянули гетерогенний центр обробки даних, який містить два типи серверів різних поколінь, як показано в таблиці 3.2.

Таблиця 3.2 – Конфігурація фізичних машин сімейства А Microsoft Azure

| Тип фізичної машині | Процесор (MIPS) | Пам'ять (MB) | Пропускна здатність (Gb/s) |
|--|-----------------|--------------|----------------------------|
| Intel Xeon 8171M 2,1 ГГц (Skylake) [x86-64] | 200 | 256 | 1 |
| Intel Xeon E5-2673 версии 3 2,4 ГГц (Haswell) [x86-64] | 420 | 512 | 2 |

Для експериментів використаємо п'ять типів віртуальних машин, як показано в таблиці 3.3. Крім того, встановимо, що всі віртуальні машини мають одну кількість ядер ЦП, і припустимо незалежність від завдань, у якій немає необхідності спілкуватися між віртуальними машинами під час виконання завдання. Щоб оцінити продуктивність VMP-ER, ми використовували різні показники, включаючи кількість активних РМ, загальне споживання електроенергії та втрату ресурсів. У нашому експерименті ми використовували реальні дані, отримані з трасування робочого навантаження PlanetLab . Для проведення експериментів використовувалася мова програмування Java.

Таблиця 3.3 – Конфігурація віртуальних машин

| Тип фізичної машині | Процесор (MIPS) | Пам'ять (МБ) | Пропускна здатність (Gb/s) |
|---------------------|-----------------|--------------|----------------------------|
| V1 | 3000 | 8000 | 4 |
| V2 | 6000 | 16000 | 8 |
| V3 | 12000 | 32000 | 12 |
| V4 | 24000 | 64000 | 16 |
| V5 | 48000 | 128000 | 20 |

Нозглянемо сценарій, у якому два типи серверів (таблиця 3.1), доступні в центрі обробки даних для розміщення віртуальних машин. Для простоти припустимо, що на цих серверах зараз доступні такі ресурси: один активний сервер має 2604 MIPS і 2800 МБ, а другий активний сервер має 3724 MIPS і 1600 МБ. А тепер давайте V1, V2, і V3 представляють віртуальні машини, які очікують розміщення в центрі обробки даних. Вимоги до ресурсів для кожної віртуальної машини такі: V1 (3000 MIPS, 8000 МБ), V2 (6000 MIPS, 16000 МБ) і V3 (12000 MIPS, 32000 МБ).

Аналізуючи різні алгоритми, було виявлено, що деякі алгоритми віддають перевагу першим серверам , а інші – другим серверам. Наприклад,

алгоритми LMBVP [19] та EEVMP [20] вибирають розміщення V1 на gthijve сервері завдяки вищому коефіцієнту енергоефективності порівняно з другим. Згодом V2 буде розміщено на сервері другого сервері, оскільки доступних ресурсів на першому сервері недостатньо. Для V3 EEVMP активує нову фізичну машину для його розміщення, оскільки ні перша, ні друга не мають відповідних ресурсів.

Навпаки, наш алгоритм враховує використання ресурсів під час визначення найбільш енергоефективного РМ серед активних РМ. Це передбачає обчислення для обох РМ з використанням нашого алгоритму, що призводить до значень 0,942 для другого та 0,861 для першого серверу. Згодом обчислюється коефіцієнт корисної дії потужності, що дає значення 218,02 для другого та 237,99 для першого. Різниця в споживанні електроенергії до і після розміщення V1 визначається, що призводить до різниці в 27,2 Вт для першого і 28,8 Вт для другого. Якщо припустити, що не відбувається порушень SLA, коефіцієнт енерго ефективності для обох серверів становить відповідно 7,55 та 7,14, що призводить до вибору першого серверу для розміщення V1.

За V2 і V3, оскільки лише другий сервер має достатньо ресурсів процесора та оперативної пам'яті, обидві віртуальні машини розміщені на другому сервері без необхідності активації іншої фізичної машини. Отже, запропонований нами алгоритм використовує менше РМ, що призводить до збільшення енергозбереження для центру обробки даних. Крім того, втрата ресурсів зведена до мінімуму, оскільки ресурси РМ використовуються належним чином. Хоча алгоритм LMBVP переміщує віртуальні машини на різні РМ під час фази міграції, це, як правило, призводить до більшого споживання енергії порівняно з запропонованим алгоритмом. Це насамперед тому, що під час міграцій споживається більше енергії. Очікується, що в центрах обробки даних із різноманітними екземплярами РМ і VM, що працюють у масштабі, наш алгоритм забезпечить кращу продуктивність.

Крім того, запропонований нами алгоритм підвищує якість обслуговування запущених РМ і мінімізує загрозу перевантаження РМ, враховуючи як поточний, так і майбутній статус використання ресурсів. Такий підхід забезпечує ефективний вибір РМ призначення. Ми враховуємо як навантаження, так і надійність РМ. Таким чином, вибір більш безпечних і збалансованих РМ для хостингу призводить до правильного використання РМ і, отже, зменшує ймовірність потенційних міграцій. Це зменшення значно покращує зниження продуктивності через метрику міграції. Збільшуючи використання ресурсів серверів, алгоритм гарантує, що менша кількість РМ зможе розмістити більше віртуальних машин. Це покращення стає життєздатним, коли кількість віртуальних машин збільшується.

Для таких великих екземплярів віртуальних машин, як Amazon EC2 M5.4xlarge (24 800 MIPS, 64 000 МБ) і M5.8xlarge (49 600 MIPS, 128 000 МБ), алгоритм LMBBP спочатку розміщує їх на серверах SR850, але пізніше переносить їх на сервери SD535, щоб мінімізувати втрату ресурсів. Цей підхід працює краще з певними типами робочих навантажень, де віртуальні машини не використовують повністю більшу частину часу (робочі навантаження не вимагають 100% своїх вимог). Однак імовірність порушення SLA збільшиться, коли ці віртуальні машини працюватимуть із повним завантаженням. Враховуючи розсіювання енергії, викликане міграцією віртуальної машини, запропонований нами алгоритм отримує кращі результати, ніж LMBBP, завдяки досягненню кращого балансування ресурсів і оптимізації енергії в середньому. Це досягається шляхом уникнення частих міграцій віртуальної машини та включення метрики SLA в процес вибору РМ, тим самим покращуючи QoS.

На рисунку 3.3 показано кількість РМ, необхідних для розміщення різної кількості віртуальних машин. VMP-ER перевершує EEVMP і LMBBP з точки зору загальної кількості активних РМ завдяки своїй ефективності упаковки, яка спрямована на виділення більшої кількості віртуальних машин одному РМ з урахуванням балансування навантаження. Навпаки, EEVMP і

LMBVP, як правило, спочатку розміщують більшість віртуальних машин на серверах SR850, оскільки їхні рішення базуються переважно на енергоефективності. У міру збільшення кількості запитів віртуальної машини обидва алгоритми починають активувати більше РМ для задоволення цих запитів, як обговорювалося раніше в цьому розділі.

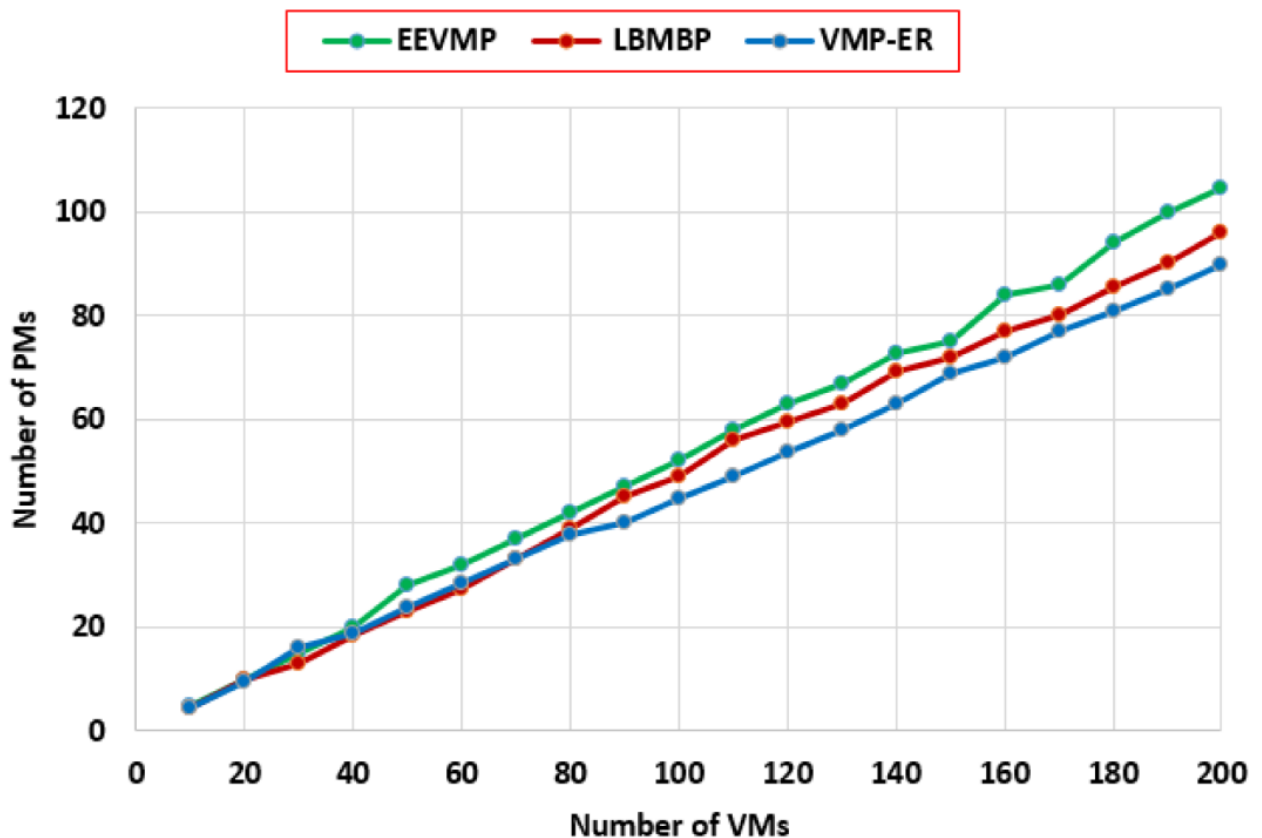


Рисунок 3.3 – Кількість фізичних машин, необхідних для розміщення вказаної кількості віртуальних машин

EEVMP вимагає більшої кількості РМ, як показано на рисунку 3.3. Це насамперед через те, що EEVMP зосереджується на найменшому збільшенні енергоспоживання під час розміщення певних віртуальних машин без урахування втрати ресурсів, що часто призводить до вибору неактивної РМ для початкового розміщення віртуальної машини. Слід зазначити, що LMBVP включає дві фази розміщення віртуальної машини: початкову фазу розміщення та фазу заміни, коли віртуальні машини переносяться на різні

PM. У нашому порівнянні ми розглядаємо лише початковий етап розміщення в оцінці.

На рисунку 3.4 порівнюється енергоспоживання для різної кількості віртуальних машин (від 10 до 200), враховуючи різні типи віртуальних машин, як показано в таблиці 3.3. На рисунку видно, що продуктивність невеликої кількості віртуальних машин однакова. Однак наш алгоритм починає випереджати EEVMP і LMBBP зі збільшенням кількості віртуальних машин.

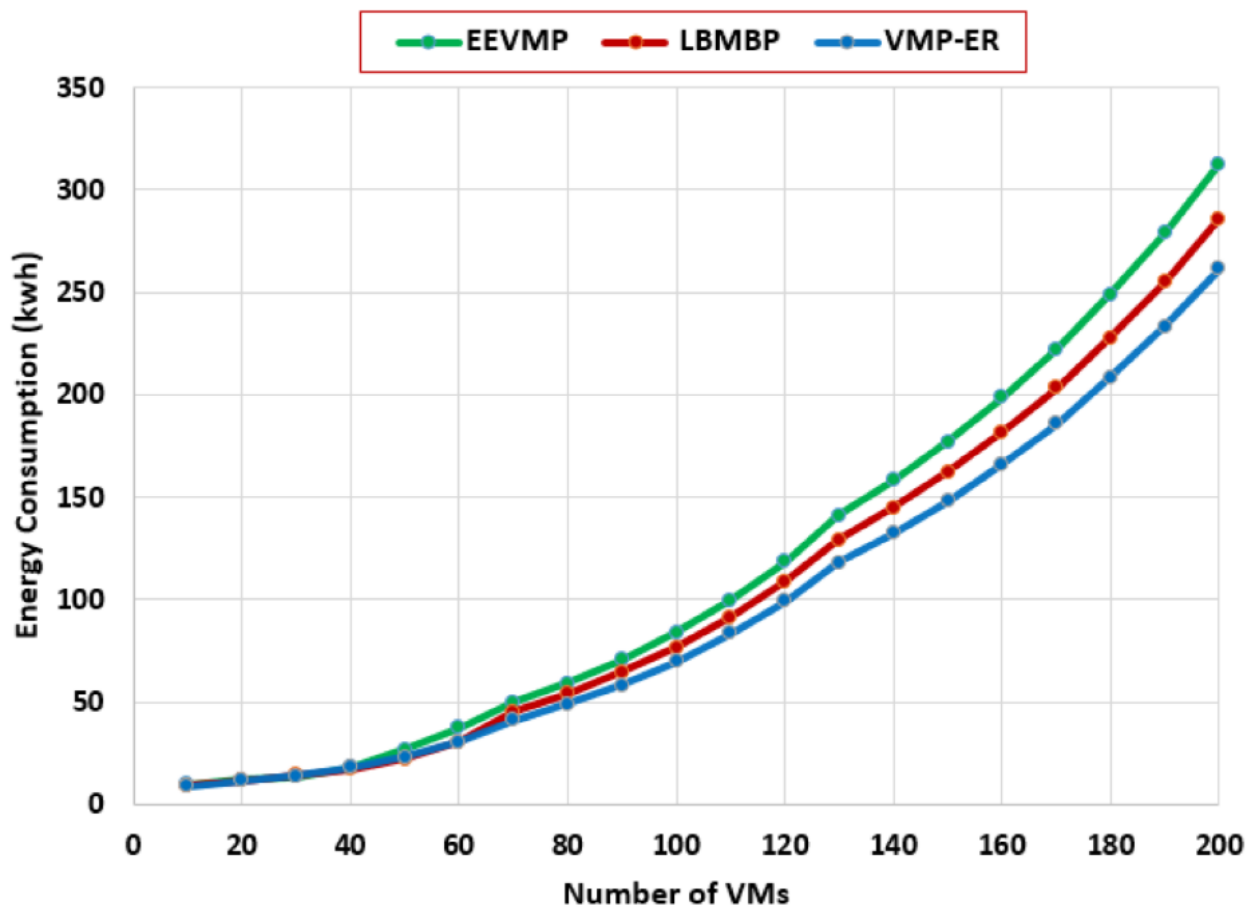


Рисунок 3.4 – Споживання енергії для заданої кількості віртуальних машин

Оскільки VMP-ER потребує менше активних PM для обробки запитів VM, це призводить до меншого споживання енергії. Крім того, збалансоване розміщення ресурсів віртуальних машин сприяє більш ефективному використанню та уникає непотрібних потенційних міграцій, що позитивно впливає на загальне енергоспоживання центру обробки даних. Навпаки, LMBBP має тенденцію до міграції віртуальних машин на другій фазі для

досягнення кращого розміщення, що, як наслідок, призводить до додаткового розсіювання енергії через міграцію. EEVMP показує більше енергоспоживання зі збільшенням кількості віртуальних машин, оскільки під час розподілу віртуальних машин алгоритм головним чином зосереджується на ресурсах центрального процесора, нехтуючи іншими ресурсами, наприклад оперативною пам'яттю. Це призводить до збільшення кількості міграцій для робочих навантажень, де віртуальні машини вимагають більшого обсягу оперативної пам'яті, що впливає на споживання енергії.

На малюнку 3.5 показано відсоток середньої втрати ресурсів (ЦП і ОЗП) при розміщенні різної кількості віртуальних машин. Зменшення кількості активних хостів свідчить про покращення середнього використання ресурсів. Для невеликої кількості віртуальних машин LMBBP працює краще, ніж EEVMP і VMP-ER.

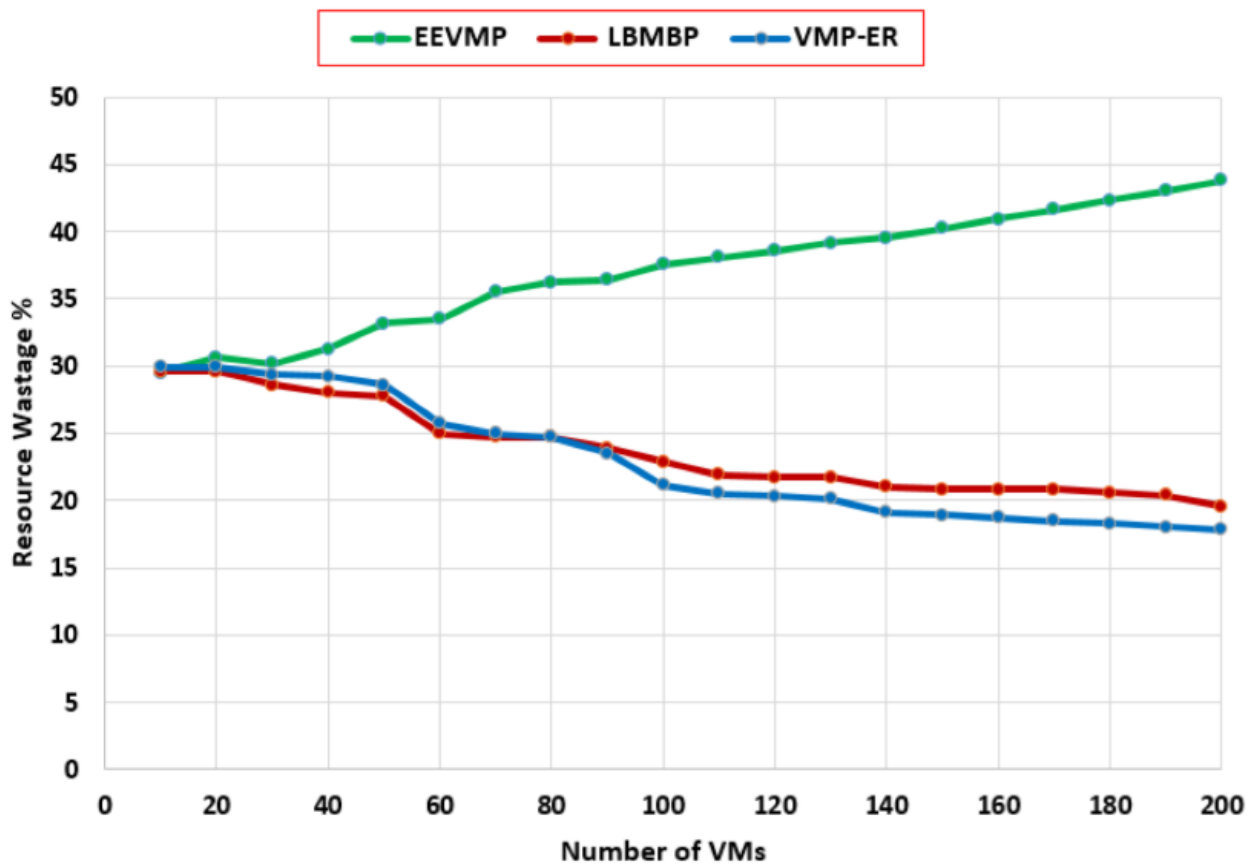


Рисунок 3.5 – Середня втрата ресурсів для певної кількості віртуальних машин

Порівнюючи продуктивність нашого алгоритму з LMBVP, обидва алгоритми ефективно використовують ресурси з мінімальною втратою ресурсів. Однак із збільшенням кількості віртуальних машин наш алгоритм починає давати кращі результати. Це пов'язано з тим, що VMP-ER підтримує більш високий рівень використання ресурсів, включаючи коефіцієнт використання ресурсів на рівні хоста, що призводить до більш ефективного вибору хоста та загальної економії енергії в центрі обробки даних.

ВИСНОВКИ

В роботі представлено комплексний огляд методів, що використовуються для вирішення проблеми розміщення віртуальної машини. Проаналізоване методи розміщення віртуальної машини на основі метаевристичних та інших алгоритмів з різними характеристиками. З цього аналізу можна зробити висновок, що більшість дослідників зосередилися на оптимізації споживання енергії хмарних центрів обробки даних. Однак деякі дослідження також зосереджено на інших важливих параметрах, таких як обчислення використання ресурсів, час виконання, час простою, параметри міграції тощо.

Основною метою методів розміщення віртуальної машини – це зменшення енергоспоживання даних без жодного впливу на рівень обслуговування користувачів хмари, що веде до збільшення прибутку від інвестицій для постачальники хмарних послуг.

Враховуючи різноманітні характеристики РМ і VM, а також багатомісні ресурси та масштабну інфраструктуру хмарного середовища, розміщення віртуальних машин стало важливою сферою досліджень. Дане дослідження вирішує цю проблему, запроваджуючи ефективний алгоритм, спрямований на мінімізацію енергоспоживання та втрати ресурсів.

Запропонований алгоритм досягає зниження енергоспоживання за рахунок зменшення кількості запущених РМ, мінімізації міграції VM, надаючи пріоритет енергоефективним РМ. Крім того, він мінімізує втрату ресурсів шляхом оптимізації розподілу ресурсів між фізичними машинами та підвищення ефективності їх використання, забезпечуючи збалансований розподіл ресурсів на рівні РМ. Часткові результати роботи опубліковано у [21].

Результати дослідження можуть бути застосовані для підвищення ефективності використання обчислювальних ресурсів у хмарних дата-

центрах, що дозволяє знизити витрати на енергоспоживання, покращити продуктивність та мінімізувати ризик перевантаження серверів.

У майбутньому треба працювати над проблемою вибору VM, коли певні PM перевантажуються. Крім того, є можливість розширити алгоритм для керування декількома ресурсами в кожному вимірі, наприклад мережею та дисковим сховищем. Також ймовірно оцінити запропонований підхід у реальному хмарному середовищі.

Майбутні дослідження можуть зосередитися на вирішенні наступних проблем. Безпека: це важливий фактор для багатьох дослідників в області віртуалізації. Зловмисники можуть отримати доступ до конфіденційних даних зі спільного пулу фізичних ресурсів у середовищі хмарних обчислень. Прогноз навантаження: точне передбачення майбутнього робочого навантаження в хмарних обчисленнях слід виконувати на основі історичних даних. Точне прогнозування робочого навантаження може зменшити міграції віртуальних машин, отже, зменшити обчислювальні витрати. Міжхмарний зв'язок: майбутні дослідження також повинні враховувати міжхмарний зв'язок під час розміщення та міграції віртуальних машин шляхом розгляду конфліктів політик різних організацій у різних хмарах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Malhotra, L., Agarwal, D., Jaiswal, A. Virtualization in cloud computing. *J Inform Tech Softw Eng*, 4.2 (2014): P. 1-3. DOI: 10.4172/2165-7866.1000136
2. Волк М.О., Мамчич О.О. Оцінювання енергетичних витрат у процесі використання мобільних пристроїв для хмарних обчислень. *Сучасний стан наукових досліджень та технологій в промисловості*. 2023. № 1 (23).– с.72-81. DOI: <https://doi.org/10.30837/ITSSI.2023.23.072>
3. Rani, Dimpi, and Rajiv Kumar Ranjan. A Comparative Study of SaaS, PaaS and IaaS in Cloud Computing. *International Journal of Advanced Research in Computer Science and Software Engineering* 4.6 (2014). P. 158-161.
4. Zoha Usmani, Shailendra Singh. A Survey of Virtual Machine Placement Techniques in a Cloud Data Center. *Procedia Computer Science*, Volume 78. 2016. Pages 491-498. <https://doi.org/10.1016/j.procs.2016.02.093>.
5. Волк М.О., Бугрій А.М., Ковтун Є.В., Брестовицький Р.М., Соробей Б.В., Лобач Я.В. Оптимізація ресурсів у хмарних обчисленнях: гібридний підхід до автоматизації операцій та енергозбереження. *Вчені записки Таврійського національного університету імені В.І. Вернадського*. Серія: Технічні науки. Том 35(74) №5 Ч.1. 2024. С. 91-96. DOI: <https://doi.org/10.32782/2663-5941/2024.5.1/15>
6. Beloglazov A, Buyya R, Lee YC, Zomaya A. A taxonomy and survey of energy-efficient data centers and cloud computing systems. In: *Advances in Computers*. Vol. 82. Elsevier. 2011. P.47-111. DOI: 10.1016/B978-0-12-385512-1.00003-7
7. Khan MA, Paplinski A, Khan AM, Murshed M, Buyya R. Dynamic virtual machine consolidation algorithms for energy-efficient cloud resource management: a review. In: *Sustainable Cloud and Energy Services*. Cham: Springer; 2018. P. 135-165. DOI: 10.1007/978-3-319-62238-5_6

8. Ahmad RW, Gani A, Hamid SHA, Shiraz M, Yousafzai A, Xia F. A survey on virtual machine migration and server consolidation frameworks for cloud data centers. *J Netw Comput Appl.* Vol. 52(10). 2015. P. 11-25. DOI: 10.1016/j.jnca.2015.02.002
9. Gabhane, J. P., Pathak, S., Thakare, N. M. Metaheuristics Algorithms for Virtual Machine Placement in Cloud Computing Environments – A Review. *Computer Networks, Big Data and IoT.* 2021. P. 329-349. DOI: 10.1007/978-981-16-0965-7_28
10. Kaushal P., Chander S., Sinha V. Virtual Machine Placement in Cloud Computing: Challenges, Research Gaps, and Future. *International Journal of Wireless and Ad Hoc Communication.* 2021. P. 64-71. DOI: 10.54216/IJWAC.030202.
11. Janani, N., R. D. Shiva, and P. Prakash. Optimization of Virtual Machine Placement in Cloud Environment Using Genetic Algorithm. *Research Journal of Applied Sciences, Engineering and Technology* Vol. 10(3). 2015. P. 274-287. DOI: 10.19026/rjaset.10.2488
12. Ivanisenko, I.,M., Volk, M.,O. (2017) Simulation methods for load balancing in distributed computing. *Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2017), Novi Sad, Serbia, September 27 – October 2.* P. 690-695. DOI: 10.1109/EWDTS.2017.8110078
13. Attaoui W., Sabir E., Multi-Criteria Virtual Machine Placement in Cloud Computing Environments: A Literature Review," 2024 International Conference on Ubiquitous Networking (UNet), Marrakech, Morocco. 2024. pp. 1-11, doi: 10.1109/UNet62310.2024.10794708.
14. Mamchych O., Volk M. A unified model and method for forecasting energy consumption in distributed computing systems based on stationary and mobile devices. *Radioelectronic and Computer Systems*, [S.l.], v. 2024, n. 2, p. 120-135. DOI: <https://doi.org/10.32620/reks.2024.2.10>.
15. Волк М.О., Бугрій А.М., Ковтун Є.В., Брестовицький Р.М., Соробей Б.В., Лобач Я.В. Оптимізація ресурсів у хмарних обчисленнях:

гібридний підхід до автоматизації операцій та енергозбереження. Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 35(74) №5 Ч.1. 2024. С. 91-96. DOI: <https://doi.org/10.32782/2663-5941/2024.5.1/15>

16. Masdari, M., Sayyid S. N., Vafa A. An overview of virtual machine placement schemes in cloud computing." *Journal of Network and Computer Applications* Vol. 66. 2016. P. 106-127. DOI: 10.1016/j.jnca.2016.01.011

17. Zhou Z., Zhigang H., Keqin Li. Virtual machine placement algorithm for both energy-awareness and sla violation reduction in cloud data centers. *Scientific Programming*. Vol. 1. 2016. P. 1-11. DOI: 10.1155/2016/5612039

18. Bai, W.; Xi, J.; Zhu, J.; Huang, S. Performance analysis of heterogeneous data centers in cloud computing using a complex queuing model. *Math. Probl. Eng.* Vol.1. 2015. No. 980945. DOI: 10.1155/2015/980945

19. Nehra, P.; Kesswani, N. Efficient resource allocation and management by using load balanced multi-dimensional bin packing heuristic in cloud data centers. *J. Supercomput.* 2023, Vol.79. P.1398–1425. DOI: 10.1007/s11227-022-04707-w

20. Sunil, S.; Patel, S. Energy-efficient virtual machine placement algorithm based on power usage. *Computing* 2023. Vol. 105. P.1597–1621. DOI: 10.1007/s00607-023-01152-2

21. Волк М.О., Бугрій А.М., Ковтун Є.В., Брестовицький Р.М., Соробей Б.В., Лобач Я.В. Оптимізація ресурсів у хмарних обчисленнях: гібридний підхід до автоматизації операцій та енергозбереження. Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 35(74) №5 Ч.1. 2024. С. 91-96. DOI: <https://doi.org/10.32782/2663-5941/2024.5.1/15>