

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 розподілу віртуальних машин в системах  
хмарних обчислень

(тема)

Виконав:

студент II курсу, групи СПМ-22-5  
Курочкін В.С.  
(прізвище, ініціали)

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. Волк М.О.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Коваленко А.А.

(прізвище, ініціали)

2024 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)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студенту \_\_\_\_\_ Курочкину Владиславу Сергійовичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Методи розподілу віртуальних машин в системах хмарних обчислень \_\_\_\_\_

затверджена наказом по університету від “ 01 ” \_\_\_\_\_ квітня \_\_\_\_\_ 2024 р. № \_\_\_\_\_ 257 Ст \_\_\_\_\_

2. Термін подання студентом роботи до екзаменаційної комісії \_\_\_\_\_ 15 червня 2024 р. \_\_\_\_\_

3. Вхідні дані до роботи \_\_\_\_\_

\_\_\_\_\_ Моделі хмарних обчислень.

\_\_\_\_\_ Постачальники хмарних послуг (AWS, GCP, Azure).

\_\_\_\_\_ Методи оцінки продуктивності хмарних платформ.

4. Перелік питань, що потрібно опрацювати у роботі \_\_\_\_\_

\_\_\_\_\_ Аналіз предметної області.

\_\_\_\_\_ Методи розподілу віртуальних машин в системах хмарних обчислень.

\_\_\_\_\_ Розробка методу розподілу віртуальних машин в системах хмарних обчислень.

\_\_\_\_\_ Експериментальна частина.

\_\_\_\_\_ Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 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	02.04.24 – 04.04.24	
2	Розробка моделей	05.04.24 – 15.04.24	
3	Реалізація алгоритмів	16.04.24 – 28.04.24	
4	Розробка структури програмних засобів	29.04.24 – 15.05.24	
5	Розробка програмних модулів	16.05.24 – 25.05.24	
6	Оформлення матеріалів кваліфікаційної роботи	26.05.24 – 05.06.24	
7	Подання кваліфікаційної роботи керівникові та попередній захист	07.06.24 – 12.06.24	
8	Подання кваліфікаційної роботи на рецензування	13.06.24 – 15.06.24	

Дата видачі завдання 01 квітня 2024 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

проф. Волк М.О.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 59 с., 11 рис., 1 табл., 1 дод., 60 джерел.

### ХМАРНІ ОБЧИСЛЕННЯ, ІНФРАСТРУКТУРА ЯК ПОСЛУГА, ГЕНЕТИЧНИЙ АЛГОРИТМ, ОПТИМІЗАЦІЯ, РОЗПОДІЛ РЕСУРСІВ.

Хмарні обчислення є найактуальнішою новою тенденцією у зберіганні, обробці, візуалізації та аналізі даних. Багато підприємств, організації та комерційних фірм використовують хмарні системи. Це стосується динамічного розподілу ресурсів на вимогу для надання гарантованих послуг клієнтам. Хмарні обчислення надають абсолютно новий підхід до надання ІТ-послуг через Інтернет. Ця парадигма дозволяє споживачам отримувати доступ до обчислювальних ресурсів, як до сервісів через Інтернет. Одною з необхідних але складних задач є розподіл ресурсів і планування у хмарних системах. В роботі розглядається гібридна стратегія для розміщення віртуальних машин на основі комбінації відомих методів. Однією з задач розробки нового методу було зменшення споживання електроенергії з покращенням ефективності. Запропонований метод можна розглядати як варіант машинного навчання для системи управління хмарними обчислюваннями. Для тестування методу використовуються трасування робочого навантаження у реальному часі. Запропонований метод показав кращі показники з точки зору використання ресурсів центру обробки даних і комп'ютерних вузлів, споживання енергії та часу виконання. Використання ресурсів, енергоспоживання та час виконання були використані як показники продуктивності в цій роботі.

## ABSTRACT

Master's thesis: 59 pages, 11 figures, 1 tables, 1 appendice, 60 sources.

CLOUD COMPUTING, INFRASTRUCTURE AS A SERVICE, GENETIC ALGORITHM, OPTIMIZATION, RESOURCE ALLOCATION.

Cloud computing is the hottest new trend in data storage, processing, visualization and analysis. Many enterprises, organizations and commercial firms use cloud systems. It refers to the dynamic allocation of resources on demand to provide guaranteed services to clients. Cloud computing provides a completely new approach to providing IT services over the Internet. This paradigm allows consumers to access computing resources as services over the Internet. One of the necessary but difficult tasks is resource allocation and planning in cloud systems. The paper considers a hybrid strategy for placing virtual machines based on a combination of known methods. One of the tasks of developing the new method was to reduce electricity consumption while improving efficiency. The proposed method can be considered as a variant of machine learning for cloud computing management system. Real-time workload traces are used to test the method. The proposed method showed better performance in terms of data center and computer node resource utilization, energy consumption, and execution time. Resource utilization, energy consumption, and execution time were used as performance metrics in this paper.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	8
ВСТУП .....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	11
1.1 Особливості хмарних обчислень .....	11
1.2 Дослідження розподілу ресурсів у хмарних системах .....	14
1.3 Характеристика хмарної архітектури .....	16
1.4 Розподіл ресурсів в хмарній системі .....	18
1.5 Основи віртуалізації в хмарних системах .....	19
2 МЕТОДИ РОЗПОДІЛУ ВІРТУАЛЬНИХ МАШИН В СИСТЕМАХ ХМАРНИХ ОБЧИСЛЕНЬ .....	24
2.1 Методи планування ресурсів .....	24
2.2 Розподіл ресурсів на основі SLA .....	25
2.3 Агентно-орієнтований підхід до розподілу ресурсів .....	25
2.4 Розподіл ресурсів на основі пріоритету .....	27
2.5 Розподіл та енергоефективність ресурсів .....	28
2.6 Адаптивне управління ресурсами .....	29
3 РОЗРОБКА МЕТОДУ РОЗПОДІЛУ ВІРТУАЛЬНИХ МАШИН В СИСТЕМАХ ХМАРНИХ ОБЧИСЛЕНЬ .....	30
3.1 Опис методу .....	30
3.2 Генетичний алгоритм .....	30
3.3 Використання методу випадкового лісу .....	33
3.4 Оцінка ефективності запропонованих рішень .....	35
4 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА .....	37
4.1 Опис середовища моделювання .....	37
4.2 Опис проведених експериментів з оцінки запропонованого методу .....	38

4.3 Опис проведених експериментів з порівнянням запропонованого методу з існуючими .....	40
4.4 Виявлені проблеми з розподілу ресурсів.....	42
ВИСНОВКИ.....	45
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	47
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	53

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- ІТ – інформаційні технології
- ПЗ – програмне забезпечення
- ПК – персональний комп'ютер
- ЦП – центральний процесор
- API – інтерфейс програмування додатків (англ., Application Programming Interface)
- AWS – Amazon Web Services
- GA – генетичний алгоритм (англ., Genetic Algorithm)
- GCP – Google Cloud Platform
- HPC – високопродуктивні обчислення (англ., High Performance Computing)
- IaaS – інфраструктура як послуга (англ., Infrastructure as a Service)
- IoT – інтернет речей (англ., Internet of Things)
- HTTP – HyperText Transfer Protocol)
- MIPS – Million Instructions Per Second
- PaaS – платформа як послуга (англ., Platform as a service)
- RA – розподіл ресурсів (англ., Resource Allocation)
- RAM – оперативна пам'ять (англ., Random Access Memory)
- RF – випадковий ліс (англ., Random forest)
- SaaS – програмне забезпечення як послуга (англ., Software as a service)
- SLA – угода про рівень обслуговування (англ., Service-Level Agreement)
- SOAP – простий протокол доступу до об'єктів (англ., Simple Object Access Protocol)
- SSD – твердотілий накопичувач (англ., Solid-State drive)
- VM – віртуальна машина (англ., Virtual Machine)
- QoS – якість обслуговування (англ., Quality of Service)

## ВСТУП

Хмарні обчислення визначаються як новий метод динамічного аналізу та підтримки адаптованих і віртуалізованих активів в Інтернеті. Kaefler [1] припустив, що хмарні обчислення – це структура, яка забезпечує доступ за вимогою до розосередженої колекції джерел, таких як хости та пристрої зберігання даних, що призводить до виконання обов'язків, які можна зберігати та видаляти з набагато меншим контролем. Хмарні обчислення пропонують передову тенденцію в генерації інформації, яка переміщує інформацію з робочого столу користувача до великих статистичних систем обробки інформації. Часто це реалізується веб-додатками згідно моделі надання як послуга.

Обчислювальні можливості в усіх хмарних застосуваннях забезпечуються колекцією центрів знань, розкиданих по всьому світу, які об'єднані високошвидкісною мережею. У хмарних обчисленнях хмара може бути розосередженою системою збору, обробки та зберегання інформації. Вона розповсюджується між клієнти з віддаленими статичними ресурсами в певний момент часу через Інтернеті. Мері [2] припустив, що підхід до керування ресурсами допомагає синхронізувати активи даних для управління роботою, виконаною як споживачами хмари, так і провайдерами. Він відомий як розподіл ресурсів (анг. Resource Allocation, RA) від постачальників до клієнтів. Оскільки адміністрування хмар дозволяє динамічно перерозподіляти завдання та дані, користувач може краще використовувати наявні потужності. Розподіл ресурсів у хмарних системах – це процес розподілу загальнодоступних ресурсів через Інтернет для відповідних хмарних програм. Інфраструктура як послуга (IaaS) надає активи запитам завдяки попередньо визначеному механізму розподілу ресурсів.

Розділ 1 описує пов'язані роботи та аналіз предметної області, розділ 2 містить подробицький аналіз сучасних методів розподілу ресурсів в хмарних

системах, розділ 3 пояснює запропонований метод, а розділ 4 описує експериментальну роботу.

Метою роботи є підвищення ефективності хмарних систем шляхом розробки метода розподілу віртуальних машин в системах хмарних обчислень.

Для реалізації мети роботи необхідно вирішити наступні задачі:

- провести аналіз сучасних хмарних систем, характеристик їх оцінювання;
- провести аналіз, порівняння та вибір методів розподілу ресурсів;
- розробити метод розподілу віртуальних машин в системах хмарних обчислень;
- провести експерименти по оцінці ефективності запропонованого методу.

Об'єктом досліджень є процес управління ресурсами в хмарних системах.

Предмет досліджень: методи розподілу віртуальних машин та завдань в системах хмарних обчислень.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Особливості хмарних обчислень

Хмарні обчислення використовують Інтернет і централізовані сервери для зберігання даних і програм. Використовуючи цей спосіб, кінцеві користувачі та організації можуть використовувати програми без введення та перевірки будь-якої конфіденційної інформації на будь-якому комп'ютері, підключеному до Інтернету. Хмарні обчислення дозволяють підвищити ефективність обчислень завдяки стандартизації даних, пам'яті, пропускну здатності та інших ресурсів. Сервіси електронної пошти, такі як Hotmail, Gmail і Yahoo, є моделями хмарних обчислень. Постачальник хмарних послуг керує всім хостом і програмним забезпеченням для керування електронною поштою в цій хмарі. Кінцевий споживач може користуватися перевагами програмного забезпечення самостійно. Хмарні обчислення служать більше як зручність, ніж як продукт, завдяки обміну ресурсами, програмами та даними між комп'ютерами та іншими пристроями.

Основними моделями надання хмарних послуг є:

- програмне забезпечення як послуга (SaaS);
- платформа як послуга (PaaS);
- інфраструктура як послуга (IaaS).

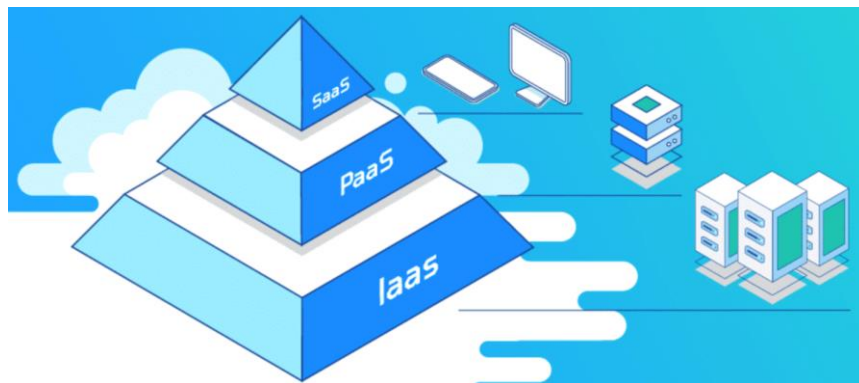


Рисунок 1.1 – Моделі надання хмарних послуг

Розподіл хмарних ресурсів має відповідати обмеженням щодо якості обслуговування (QoS), наданим клієнтами через угоди про рівень обслуговування (SLA), а також має зменшити споживання енергії.

Nair та ін. [3] припустив, що для того, щоб виконувати хмарні сервіси максимально ефективно та результативно, необхідно розподілити ресурси. Крім того, його можна розглядати як будь-який процес, який гарантує, що інфраструктура постачальника послуг може відповідати визначеним вимогам SLA. Для того, щоб задовольнити вимоги хмарних додатків еластичним і прозорим способом, розподіл ресурсів визначається як процес інтеграції діяльності хмарних провайдерів з метою використання та розподілу обмежених ресурсів, які можуть здаватися користувачам безмежними [3]. Дві ключові сторони в хмарних обчисленнях (користувачі та постачальники послуг) отримують вигоду від схем розподілу ресурсів. Основними трендами хмарних обчислень є якість та надійність. Окрім цього користувачі можуть захотіти передбачити своє робоче навантаження, щоб завершити проект достроково. З іншого боку, це може призвести до проблеми з надлишковим забезпеченням. З іншого боку, постачальники хочуть оптимізувати свій прибуток, використовуючи менше ресурсів на користувача, щоб задовольнити більше клієнтів і отримати більший дохід. Результатом буде брак ресурсів. Однак відсутність обміну інформацією між ними унеможливує розподіл ресурсів у взаємовигідний спосіб.

В роботі [4] описано, як приваблива обчислювальна модель хмари приваблюють багатьох клієнтів, проектів та інтернет-організації. Основною метою як постачальників хмарних послуг, так і споживачів хмарних послуг є максимізація хмарних ресурсів і максимізація фінансового прибутку. Серед найбільш нагальних проблем хмарних обчислень, здається, розподіл обчислювальних ресурсів. Уявлення споживачів про розподіл ресурсів формується тим, як продукти та послуги поширювалися серед таких груп населення. Чим продуктивніша економіка, тим більше ресурсів їй знадобиться для роботи.

Клієнти отримують доступ до ресурсів, необхідних для ефективної роботи, використовуючи навички як послугу. Це позбавляє клієнтів від необхідності платити за ресурси, які ніколи не використовуються. Компанії можуть модернізувати існуючі операції, дозволивши клієнтам отримати доступ до найновіших програм і змін інфраструктури. Щоразу, коли дана цільова функція потребує найкращої відповідності ресурсу та роботи за місцем і часом, розподіл ресурсів і планування в розподілених системах відіграють вирішальну роль у визначенні найкращої відповідності.

Враховуючи складність найкращого розподілу ресурсів, тобто ефективного розподілу з дефіцитними ресурсами та найбільшим прибутком, розподіл ресурсів між користувачами хмари здається складною процедурою. Програми робочого процесу можуть бути реалізовані швидше завдяки динамічному розподілу ресурсів, що дозволяє клієнтам визначати свої політики. Реплікація розподілу ресурсів інфраструктури хмарних обчислень виглядає так, що кілька ресурсів із такої уніфікованої групи ресурсів були розподілені одночасно.

Зі збільшенням кількості користувачів комп'ютерів і мобільних пристроїв за останні роки збереження даних стало пріоритетом у всіх сферах. Великі та малі підприємства сьогодні покладаються на дані, і вони витрачають багато грошей, щоб підтримувати ці дані в актуальному стані. Хмарне сховище робить ІТ-послуги доступними на вимогу через великі розподілені центри обробки даних і високошвидкісні мережі. Віртуалізація мережі — це нещодавня тенденція в хмарних обчисленнях, яка з'явилася як багатогранний підхід до майбутнього Інтернету завдяки полегшенню спільних ресурсів. Надання віртуальної мережі розглядається як значний виклик з точки зору створення NP-складних проблем, мінімізації часу обробки робочого процесу під контрольованими ресурсами тощо.

Робота з розподілом ресурсів і плануванням у хмарних обчисленнях і необхідна, і складна. У цій роботі метод машинного навчання "випадковий ліс" (анг. Random forest, RF) і генетичний алгоритм (анг. Genetic Algorithm,

GA) об'єднані в гібридну стратегію для розміщення віртуальних машин. Це приклад керованого машинного навчання. Метою проекту є максимізація використання ресурсів шляхом мінімізації енергоспоживання при кращому розподілі та використанні ресурсів. Існує метод створення навчальних даних, які можна використовувати для навчання випадкового лісу. Метод тестується за допомогою трасування робочого навантаження у реальному часі. Робочий процес організовано, як показано нижче.

## 1.2 Дослідження розподілу ресурсів у хмарних системах

Було проведено кілька досліджень щодо аналізу розподілу ресурсів хмарної платформи. Алгоритми для розподілу ресурсів були випробувані з використанням різних методів планування. Під час розподілу ресурсів у хмарі враховуються численні фактори, такі як максимальна пропускна здатність, висока продуктивність, якість обслуговування та виконання SLA. Нижче наведено огляд досліджень, які безпосередньо стосуються розподілу ресурсів.

В роботі [5] автори для досягнення оптимізації хмарного планування пропонують вдосконалену техніку планування ресурсів, засновану на ретельному дослідженні хмарних систем IaaS (інфраструктура як послуга). Гнучкий метод поширення віртуальних машин досліджується для найкращого використання фізичних ресурсів. Політика автоматизованого розвитку була реалізована за допомогою IGA.

Робота [6] використовує ринкову стратегію розподілу ресурсів (RAS-M) для зменшення кількості ресурсів, споживаних великими центрами обробки даних, одночасно надаючи послуги кращої якості хмарним клієнтам. У рамках хмарних обчислень ресурси можна гнучко розподіляти на вимогу.

Наступне припущення, розглянуте у роботі [7] засноване на том, що угоди про рівень обслуговування (SLA) регулюють надання хмарних послуг. Необхідно постійно контролювати порушення SLA для мінімізації вартості.

Існує кілька критеріїв SLA, які слід перевірити під час розподілу послуг. В даній роботі було оцінено механізм планування з кількома параметрами SLA. Відповідно трьом типам потоків планування завдань у хмарах розглядаються три приклади моделей обслуговування, що розроблено для інфраструктури, платформи та програмного забезпечення. Можна надати безпечний доступ до інформації кільком незалежним клієнтам без будь-яких додаткових витрат на обслуговування. Основна проблема хмарних обчислень полягає в тому, щоб клієнти отримували максимум від ресурсів, які вони придбали, враховуючи їхні обмеження на обчислення. Якщо цю проблему не вирішити, хмарна система в цілому може бути неефективною. Стратегія спрямована на динамічний розподіл ресурсів і найкраще їх використання протягом встановленого періоду [7]. Пропонується модель розподілу ресурсів на основі правил (RBRAM), яка розподіляє ресурси відповідно до важливості завдання, щоб уникнути неефективності та використання ресурсів. Через Інтернет можна орендувати обчислювальні ресурси за допомогою хмарних обчислень, відомих як IaaS. Замовник може вибрати з різних комп'ютерних ресурсів, виходячи зі своїх особливостей та вимог. Концепція IaaS була використана для розподілу ресурсів на виконання всіх завдань. Існують обмежені терміни виконання проектів Real-Time.

Використовуючи цей підхід, обчислювальним і мережевим ресурсам надається однакова вага в рішеннях щодо розподілу ресурсів. В роботі [8] пропонується ідея, що федеративна обчислювальна та мережева система (FCN) є новою технологією, призначеною для обробки цього спільного розподілу. Проблема перевантаження виникає щоразу, коли запит на послугу вимагає розподілу різноманітних ресурсів (наприклад, обчислювальної потужності, сервера зберігання та підключення).

Для ефективного спільного використання ресурсів у перевантаженому сценарії, в роботі [9] представили механізм контролю перевантаження. Оскільки різні ресурси виділяються одночасно для задоволення потреб у кожній службі, цей метод можна розглядати як ефективне управління

доступними ресурсами. Це вимога служб хмарних обчислень, які розподіляють пропускну здатність, щоб одночасно вводити в методи розподілу.

В роботі [10] запропоновано об'єднання ресурсів в систему хмарних обчислень з обмеженням на потужність електроенергії. Одночасно враховуються як здатність адміністрування, так і пропускну здатність платформи. Декілька ресурсів із такої загальної групи ресурсів одночасно призначаються кожному пошуку для певного віку відповідно до методу розподілу ресурсів для такого середовища хмарних служб. Тут оцінюється здатність до забезпечення SLA та спроможність ефективного використання обчислювальних ресурсів.

Як наслідок, ця частина містить різні дослідження, пов'язані з розподілом ресурсів хмарних обчислень IaaS. Хоча було запропоновано численні рішення щодо планування, розподілу QoS ресурсів, непередбачуваності, споживання енергії та інші проблеми приводять до ситуації, що жодне з них не могло досягти розподілу ресурсів у реальному часі. Нами запропоновано метод, який базується на алгоритмах машинного навчання на основі підкріпленого навчання, які допомагають нам отримати підвищення ефективності щодо розподілу ресурсів.

### 1.3 Характеристика хмарної архітектури

Архітектуру хмарних обчислень можна розділити на три рівні, як показано на рисунку 1.2. Існує три типи ресурсів:

- ресурс;
- платформа;
- програма.

Перший рівень, рівень ресурсів (рівень інфраструктури), також складався з двох характеристик. По-перше, це обчислення на фізичній та

віртуалізованій основі. По-друге, мережа, ресурси зберігання та пристрої пам'яті, запропоновані Mochizuki та Kuribayashi [11].

Наприклад, клієнти можуть платити лише за розділ, з яким вони взаємодіють у сховищі, замість того, щоб придбати диск або нічого не знати про область запису [12].

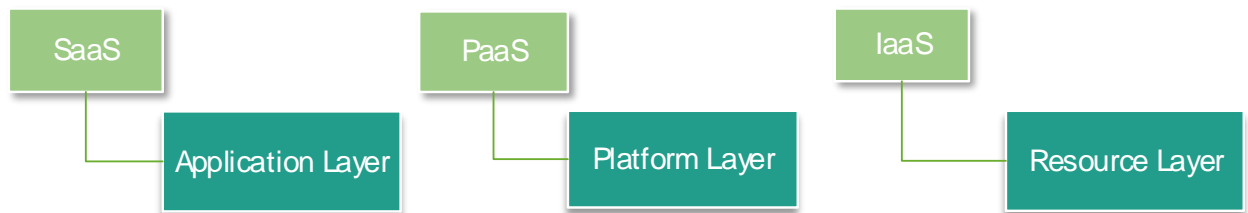


Рисунок 1.2 – Архітектура хмарних послуг

У роботі [13] зроблено припущення, що прикладний рівень (також відомий як програмне забезпечення як послуга) відновлює функції системи. Якщо ви використовуєте SaaS, вам не доведеться встановлювати та запускати великий пакет програмного забезпечення на вашому комп'ютері. Використання зразка з оплатою за використання замість придбання дорогого програмного продукту може допомогти вам заощадити гроші. У хмарному шаблоні є чотири моделі розгортання.

**Приватна хмара:** організації рідко використовують хмарну інфраструктуру, якою може керувати або компанія, або третя сторона. Вона може бути локально або розміщена віддалено.

**Хмара спільноти:** у цій хмарі інфраструктура спільно використовується декількома фірмами, допомагаючи певній групі людей, які мають подібні задачі. Місія, стандарти безпеки, політика та питання відповідності – це лише деякі з них. Такою хмарою можуть керувати окрема фірма, державні органи або третя сторона. Її можна зустріти як на території однієї з фірм, так і поза нею.

**Громадська хмара:** отримання права власності на хмару через

корпорацію, яка надає хмарні послуги. Вона доступний як для населення, так і для великих галузевих команд.

Гібридна хмара: кожна публічна та приватна хмара має статут гібридної хмари. Її можуть формувати різні компанії, але вони пов'язані разом контрольованими процесами, які дозволяють передавати записи та програмне забезпечення.

#### 1.4 Розподіл ресурсів в хмарній системі

Як було сказано раніше, розподіл ресурсів – це метод призначення оцінюваних ресурсів необхідним хмарним додаткам у хмарних системах через Інтернет. Незважаючи на те, що обмежені ресурси є спільними, однією з найактуальніших проблем у хмарних обчисленнях є розподіл ресурсів. Розподіл ресурсів стосується розподілу послуг між клієнтами з точки зору споживача. Переваги розподілу ресурсів включають той факт, що користувачеві не потрібно встановлювати апаратне чи програмне забезпечення для доступу, розробки та зберігання програми через Інтернет. Немає обмежень щодо того, куди ви можете йти або як ви можете туди дістатися. Клієнт може отримати доступ до даних і програми з будь-якого місця та на будь-якій платформі у всесвіті. Хмарні провайдери поширювали свої ресурси по всьому Інтернету в часи дефіциту. Далі ми розглянемо чотири різні способи використання лімітів обчислення хмарного провайдера:

- попереднє резервування: у цьому випадку зарезервовані активи мають бути завжди доступними;
- максимальна кількість зусиль: якнайшвидше відновлення активів за даних обставин виконується на рівні всіх засобів; запит клієнта може бути прийнятий або відхилений на основі доступних ресурсів; Якщо запит клієнта прийнято, ресурси надаються йому негайно;
- чутливий дедлайн: коли планування може гарантувати, що віно буде завершене раніше встановленого терміну.

## 1.5 Основи віртуалізації в хмарних системах

Віртуалізація є основоположною технологією, застосовною як для впровадження хмарних обчислень, так і для великих даних. Вона забезпечує основу для багатьох атрибутів платформи, необхідних для доступу, зберігання, аналізу та керування розподіленими обчислювальними компонентами в середовищах великих даних.

Віртуалізація – процес використання комп’ютерних ресурсів для імітації інших ресурсів – цінується за її здатність підвищувати використання ІТ-ресурсів, ефективність і масштабованість. Одним із основних застосувань віртуалізації є консолідація серверів, яка допомагає організаціям збільшити використання фізичних серверів і потенційно заощадити витрати на інфраструктуру. Однак віртуалізація має багато переваг. Компанії, які спочатку зосереджувалися виключно на віртуалізації серверів, тепер усвідомлюють, що її можна застосувати в усій ІТ-інфраструктурі, включаючи програмне забезпечення, сховища та мережі.

Віртуалізація відокремлює ресурси та служби від основного фізичного середовища, що дозволяє створювати багато віртуальних систем в одній фізичній системі. На рисунку 1.3 показано типове середовище віртуалізації.

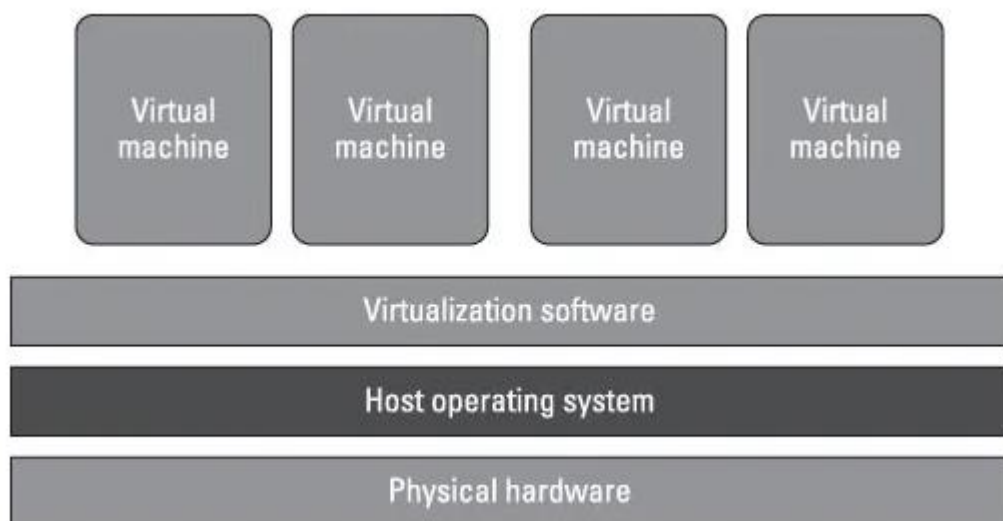


Рисунок 1.3 – Типове середовище віртуалізації

Однією з головних причин, чому компанії впроваджують віртуалізацію, є підвищення продуктивності та ефективності обробки різноманітних робочих навантажень. Замість того, щоб призначати виділений набір фізичних ресурсів для кожного набору завдань, об'єднаний набір віртуальних ресурсів можна швидко розподілити за потреби для всіх робочих навантажень. Покладення на пул віртуальних ресурсів дозволяє компаніям зменшити затримку. Це збільшення швидкості та ефективності надання послуг є функцією розподіленого характеру віртуалізованих середовищ і допомагає покращити загальний час до окупності.

Використання розподіленого набору фізичних ресурсів, таких як сервери, більш гнучким і ефективним способом забезпечує значні переваги з точки зору економії коштів і підвищення продуктивності. Ця практика має кілька переваг, зокрема:

- віртуалізація фізичних ресурсів (таких як сервери, сховища та мережі) дозволяє істотно покращити використання цих ресурсів;
- віртуалізація дозволяє покращити контроль над використанням і продуктивністю ваших ІТ-ресурсів;
- віртуалізація може забезпечити певний рівень автоматизації та стандартизації для оптимізації вашого обчислювального середовища;
- віртуалізація забезпечує основу для хмарних обчислень.

Незважаючи на те, що можливість віртуалізації ресурсів значно підвищує ефективність, це не обходиться без витрат. Віртуальними ресурсами потрібно керувати так, щоб вони були безпечними. Наприклад обробка зображення може стати технікою для зловмисника, щоб отримати прямий доступ до критичних систем. Крім того, якщо компанії не мають процесу видалення невикористаних зображень, система більше не працюватиме ефективно.

Важливість віртуалізації для великих даних. Вирішення проблем із великими даними зазвичай потребує керування великими обсягами розподілених сховищ даних разом із використанням програм, що потребують

інтенсивних обчислень і даних. Тому потрібне високоефективне ІТ-середовище для підтримки великих даних.

Віртуалізація забезпечує додатковий рівень ефективності, щоб зробити платформи великих даних реальністю. Хоча віртуалізація технічно не є обов'язковою вимогою для аналізу великих даних, програмні інфраструктури, такі як MapReduce, які використовуються в середовищах великих даних, більш ефективні у віртуалізованому середовищі. Якщо потрібно масштабувати середовище великих даних – майже без обмежень – слід віртуалізувати елементи середовища. Віртуалізація має три характеристики, які підтримують масштабованість і ефективність роботи, необхідні для середовищ великих даних:

- розподіл: у віртуалізації багато програм і операційних систем підтримуються в одній фізичній системі шляхом розподілу (розділення) доступних ресурсів;

- ізоляція: кожна віртуальна машина ізолювана від фізичної системи хоста та інших віртуалізованих машин; завдяки цій ізоляції, якщо один віртуальний екземпляр аварійно завершує роботу, це не впливає на інші віртуальні машини та хост-систему;

- інкапсуляція: віртуальну машину можна представити (і навіть зберегти) як один файл, тож ви можете легко ідентифікувати її на основі послуг, які вона надає.

Наприклад, файл, що містить інкапсульований процес, може бути повною бізнес-службою. Ця інкапсульована віртуальна машина може бути представлена як повна сутність. Таким чином, інкапсуляція може захистити кожен програму, щоб вона не заважала іншим програмам.

Однією з найважливіших вимог для успіху роботи з великими даними є належний рівень продуктивності для підтримки аналізу великих обсягів і різноманітних типів даних. Коли використовуються такі середовища, як Hadoop і MapReduce, дуже важливо, щоб була допоміжна інфраструктура, яка може масштабуватися. Віртуалізація підвищує ефективність на кожному

рівні IT-інфраструктури. Застосування віртуалізації у конкретному середовищі допоможе досягти масштабованості, необхідної для аналізу великих даних.

Впровадження віртуалізації за допомогою наскрізного підходу забезпечить переваги для великих даних та інших типів робочих навантажень у вашому середовищі. Наскрізний підхід означатиме, що помилки можна виправляти швидше – вимога в середовищі великих даних. Під час роботи з великими даними інфраструктура має бути готова керувати потенційно дуже великими (об'єм), дуже швидкими (швидкість) і дуже неструктурованими (різноманітність) даними.

У результаті все IT-середовище потрібно оптимізувати на кожному рівні, від мережі до баз даних, сховищ і серверів. Якщо віртуалізуються лише свої сервери, можна зіткнутися з вузькими місцями через інші елементи інфраструктури, такі як сховище та мережі. Якщо зосередитися лише на віртуалізації одного елемента інфраструктури, з меншою ймовірністю будуть досягнуті необхідні затримки та ефективність, а також, швидше за все, компанія зазнає вищі витрати і ризики для безпеки.

Реальність така, що більшість організацій не намагаються віртуалізувати всі елементи своєї інфраструктури одночасно. Багато організацій починають з віртуалізації серверів і досягають певного рівня підвищення ефективності. Реально інші елементи можуть бути віртуалізовані, якщо це необхідно, щоб продовжувати покращувати загальну продуктивність і ефективність системи.

При віртуалізації серверів один фізичний сервер розділений на кілька віртуальних серверів. Апаратне забезпечення та ресурси машини – включаючи оперативну пам'ять (RAM), ЦП, жорсткий диск і мережевий контролер – можна віртуалізувати (логічно розділити) на ряд віртуальних машин, кожна з яких запускає власні програми та операційну систему. Віртуальна машина (VM) – це програмне представлення фізичної машини, яка може виконувати або виконувати ті самі функції, що й фізична машина.

Тонкий шар програмного забезпечення фактично вставляється в апаратне забезпечення, яке містить монітор віртуальної машини або гіпервізор. Гіпервізор можна розглядати як технологію, яка керує трафіком між віртуальними машинами та фізичною машиною.

Віртуалізація сервера використовує гіпервізор для забезпечення ефективності використання фізичних ресурсів. Звичайно, встановлення, налаштування та адміністративні завдання пов'язані з налаштуванням цих віртуальних машин. Це включає в себе керування ліцензіями, керування мережею та адміністрування робочого навантаження, а також планування потужності.

Віртуалізація сервера допомагає гарантувати, що ваша платформа може масштабуватися за потреби для обробки великих обсягів і різноманітних типів даних, включених до аналізу великих даних. Можливо не знати обсяг чи різноманітність структурованих і неструктурованих даних, необхідних перед початком аналізу. Ця невизначеність робить потребу у віртуалізації серверів ще більшою, надаючи середовищу можливість задовольнити неочікуваний попит на обробку дуже великих наборів даних.

Крім того, віртуалізація серверів забезпечує основу, яка дозволяє використовувати багато хмарних служб як джерела даних для аналізу великих даних. Віртуалізація підвищує ефективність хмари, що полегшує оптимізацію багатьох складних систем. У результаті організації мають продуктивність і оптимізацію, щоб мати доступ до даних, які раніше були недоступні або дуже важко зібрати. Платформи великих даних все частіше використовуються як джерела величезної кількості даних про вподобання, настрої та поведінку клієнтів. Компанії можуть інтегрувати цю інформацію з внутрішніми даними про продажі та продукцію, щоб отримати уявлення про вподобання клієнтів і робити більш цілеспрямовані та персоналізовані пропозиції.

## 2 МЕТОДИ РОЗПОДІЛУ ВІРТУАЛЬНИХ МАШИН В СИСТЕМАХ ХМАРНИХ ОБЧИСЛЕНЬ

### 2.1 Методи планування ресурсів

Техніки розподілу ресурсів у хмарному середовищі можуть використовуватися різними способами. Експерти в цій галузі постійно розвиваються та винаходять нові методи, щоб відповідати вимогам часу. Навіть якщо нові технології очікуються щодня, є кілька, яких неможливо уникнути і чий внесок є критично важливим. Ми розглянемо кілька методів, які показано на рисунку 2.1.



Рисунок 2.1 – Методи розподілу ресурсів у хмарних системах

В роботі [14] детально обговорено план резервування та плани на вимогу користувача. Порівняно з ціною плану на вимогу, ціна плану

резервування має бути нижчою. Бо замовник уже погодився сплатити суму. Однак через невизначеність щодо ціноутворення ресурсів постачальника не згадується майбутня ціна попиту користувача. Було запропоновано новий метод, який включає довгостроковий план у рішення. Цей автор пропонує створення кооперативних хмарних ринків для розподілу ресурсів. Мета полягає в тому, щоб знизити витрати, водночас забезпечуючи споживача необхідними ресурсами. За допомогою цього алгоритму можна зменшити витрати. В роботі [15] розглянуті основи ринково-орієнтованої хмари.

## 2.2 Розподіл ресурсів на основі SLA

За допомогою цього методу користувач і постачальник послуг узгоджують угоду про рівень обслуговування. SLA має відповідати потребам кінцевого користувача. У роботі [16] представлена модель розподілу ресурсів на основі SLA. Було досліджено кілька постачальників хмарних послуг. Поняття рівноваги Неша в теорії ігор використовується для розподілу ресурсів між споживачами в цій стратегії. Розподіл ресурсів постачальників мультимарних послуг – це гра. Для кожного CSP є сервери, і запити маршрутизуються відповідно.

Робота [17] досліджує, як реорганізувати ресурси. Перерозподіл ресурсів, запропонований автором, розбито на дві частини. Контракт SLA та планування ресурсів обробляються центром обробки даних і клієнтом відповідно в першій частині, а контрольна діаграма, яка створюється у другій частині, виявляє порушення SLA з урахуванням продуктивності хоста. Оцінка базується на використанні функції корисності.

## 2.3 Агентно-орієнтований підхід до розподілу ресурсів

Останнім часом було проведено кілька досліджень на тему оптимізації хмарних ресурсів. Що стосується оптимізації хмарних ресурсів, робота [18]

містить рішення, засноване на зменшенні карти пошуку [19]. PSO займається здатністю зграї птахів летіти до спільної мети, одночасно проводячи власні незалежні пошуки в процесі. Кожна частинка може спілкуватися одна з одною та оновлювати своє нове розташування. Кожна частинка має унікальну користь, але є й універсальна. Частинка рухається з певною швидкістю. Існує три частини структури оптимізації хмарних ресурсів: час, гроші та задоволення кінцевого користувача. При визначенні обсягу роботи та розподілу ресурсів використовується запропонований алгоритм.

Метод мурашиних колоній був використаний в роботі [20] для призначення активів споживачам. Стратегія пошуку їжі мурахи визначає ідею цього алгоритму. Алгоритм, натхненний мурахами, використовує методи вирішення NP, щоб імітувати поведінку диких мурах у пошуках їжі. За допомогою цього підходу шляхи також можуть бути якомога коротшими. Алгоритм ACO виконується на роботі, яку робочий агент збирає від клієнта. Було показано, що феромони мають вагу, яка використовується для спілкування. Метод Ant colony вибирає вузли для мережі. Ця політика регулює розподіл хмарних ресурсів.

Оптимізація плодової мушки (PFOA) була запропонована в роботі [20] для розподілу ресурсів і організації завдань у хмарі. Щоб привернути увагу широкої громадськості, проводиться відповідний аналіз.

У роботі [21] удосконалено метод k-середніх на основі пріоритетів для розподілу завдань. Якщо покращене значення недоступне, пошук продовжується. Розподіл ресурсів можна зробити більш ефективним, зменшивши витрати. Автори роботи [22] представили новий алгоритм оптимізації для того, щоб зменшити витрати на впровадження, задовольнивши вимоги клієнтів. Методика розроблена для використання в хмарному центрі обробки даних для створення веб-додатків.

Хмарний провайдер і веб-додатки використовуються як вхідні дані. За допомогою алгоритму хмарні ресурси можна консолідувати найефективнішим способом. Генетичний алгоритм використовувався для

прогнозування ресурсів хмарного центру обробки даних. Виходячи з поточної статистики, очікується, що будуть доступні додаткові ресурси. Для завершення прогнозування використовується генетичний алгоритм. Вихідні дані прогнозування використовуються для призначення віртуальної машини.

#### 2.4 Розподіл ресурсів на основі пріоритету

В роботі [23] автори запропонували підхід до виконання завдань на базі пріоритетів. Крім того, обговорюється ідея повторного використання віртуальних машин в наступних завданнях. Обслуговування нового завдання не потребує створення нових віртуальних машин. Виконати високопріоритетне завдання можна, призупинивши низькопріоритетне. Якщо віртуальна машина виконала єдине завдання, призупинене завдання може бути знову відкрито на цій віртуальній машині. Механізм розподілу ресурсів представлено в роботі [24], в якій передбачається однорангове хмарне обчислення. Завдання з більшою обчислювальною потужністю мають пріоритет над іншими завданнями. Класи пріоритетів, такі як високий, середній і низький, створюються за допомогою К алгоритму, а завдання в черзі сортується відповідно.

Вартість ресурсу постійно і безперервно оцінюється за запропонованою методикою на основі поточного попиту на цей ресурс. В Інтернеті проводиться багато дискусій щодо впровадження активів з однорангових хмар. У роботі [25] представлено механізм призначення ресурсів із відмовостійкістю. Виконання починається від завдання з найнижчим пріоритетом до завдання з найвищим пріоритетом. У цьому процесі використовується більш складний механізм резервування. За допомогою цього методу створюється черга завдань на основі пріоритетів, а потім призначаються пріоритетні завдання. Для завдань з високим пріоритетом резервуються більш потужні ресурси. Крім того, цей підхід можна

використовувати для забезпечення функціональної стійкості до відмов хмарних програм.

## 2.5 Розподіл та енергоефективність ресурсів

Робота [24] припонує «техніку аукціону», яка реалізує стратегією розподілу ресурсів та використовується в ринково-орієнтованій хмарі. У даному методі ресурси купуються та продаються на основі вподобань або вимог користувача. Основними функціями цього підходу є визначення оптимальної ціни та визначення переможця аукціону. Portable Format for Analytics (PFA) – це інструмент для визначення ідеального покупця та продавця. Дослідження було зосереджено на вподобаннях учасників аукціону щодо продажу та купівлі ресурсів за ціною, яка їх влаштовує. У цьому випадку використовуються комбінаторні аукціони (беруть участь як споживачі, так і постачальники). Нечесних учасників аукціону можна виявити та усунути за допомогою ідентифікаторів. Система відповідає за надання високопродуктивних обчислювальних ресурсів.

Використання політики розділу та заміни віртуальних машин запропонована в статті [26]. Наявні ресурси розподіляються через ідентифікацію перевантажених ресурсів. Споживання енергії за допомогою неактивного сервера призначається для запиту на вимогу шляхом вибору та заміни віртуальної машини. Метод дозволяє економити енергію, споживану серверами.

Є ряд досліджень споживання енергії у мобільних хмарах. Ця тема була предметом численних досліджень [27]. Дослідження спрямовано на зменшення споживання енергії в хмарних центрах обробки даних. Запропонована структура досягає наступних цілей. На початку визначається кількість необхідних ресурсів і кількість запитів віртуальної машини, які будуть зроблені з боку користувачів. Розраховано, скільки фізичних машин (PM) потрібно для задоволення потреб користувачів. Метод зменшує

кількість енергії, споживаної хмарними центрами обробки даних. Ця система включає класифікацію даних, прогнозування робочого навантаження та керування живленням.

Стаття [28] містить опис стратегії розподілу ресурсів при одночасному зниженні споживання енергії. За допомогою цієї стратегії можна зробити міграцію віртуальних машин більш ефективнішою. Використання ЦП і міграція віртуальної машини плануються на основі часу виконання.

Розроблений авторами метод прогнозування серій в роботі [29], засновано на моделі використання хмари IaaS. Метод використовує дві моделі: одну для оптимізації потужності сервера, а іншу для зменшення втрати ресурсів сервера в хмарі IaaS за допомогою методології, яка вимірює витрати. Запити віртуальної машини від різних користувачів із різними вимогами до ресурсів були враховані для моделі втрати ресурсів, тоді як споживання енергії на сервері враховувалося для моделі потужності сервера.

## 2.6 Адаптивне управління ресурсами

У системі Cloud Computing система управління програмами та ресурсами була створена за допомогою адаптивної стратегії управління, відомої як «невідомість». У статті розглядається стратегія оцінки невідомості та застосування приватних хмар. Для обробки непередбачуваного робочого навантаження використовується автономна адаптивна структура, яка кваліфікує або дискваліфікує довільні сектори, такі як компоненти програми або служби. Щоб залишатися гнучким, був розроблений механізм відключення. Життєдіяльність моделі можна забезпечити шляхом тимчасового відключення деяких програм. Можна збільшити швидкість прийняття запитів, дезактивувавши певні довільні дії та повторно використовуючи ці ресурси для більш важливих програм. Крім того, відмова від роботи дозволяє збільшити просування ресурсів, зберігаючи функціональність програм, щоб уникнути великих навантажень.

## 3 РОЗРОБКА МЕТОДУ РОЗПОДІЛУ ВІРТУАЛЬНИХ МАШИН В СИСТЕМАХ ХМАРНИХ ОБЧИСЛЕНЬ

### 3.1 Опис методу

Запропонований у даній роботі метод розміщення віртуальної машини базується на генетичному алгоритмі та алгоритмі випадкового лісу. Випадковий ліс навчається за допомогою набору даних, створеного генетичним алгоритмом, який реалізує техніку оптимізації. Віртуальні машини призначаються фізичним комп'ютерам у наборі даних з метою навчання. Пропонується гібридна модель, яка заснована на генетичному алгоритмі та методі випадкового лісу для розміщення віртуальної машини. Генетичний алгоритм – це техніка оптимізації, яка генерує набір даних, необхідний для навчання алгоритму випадкового лісу. Навчальний набір даних складається з розподілу віртуальної машини або зіставлення з фізичними машинами.

### 3.2 Генетичний алгоритм

Одним із найпопулярніших методів пошуку найкращої відповіді є генетичний алгоритм (GA). При його застосуванні відображення віртуальних на фізичні машини  $VM \rightarrow PM$  генерується генетичним алгоритмом. Наш метод включає наступні кроки, як показано на рисунку 3.1. Алгоритм складається з наступних кроків.

Крок 1: ініціалізація множин завдань та ресурсів.

Крок 2: обчислення функції допасованості

$$Fs(i) = \beta * PE, \quad (3.1)$$

де  $i$  – характеризує особу (варіант розподілу) в популяції,  $PE$  – енергоефективність обраної особи.

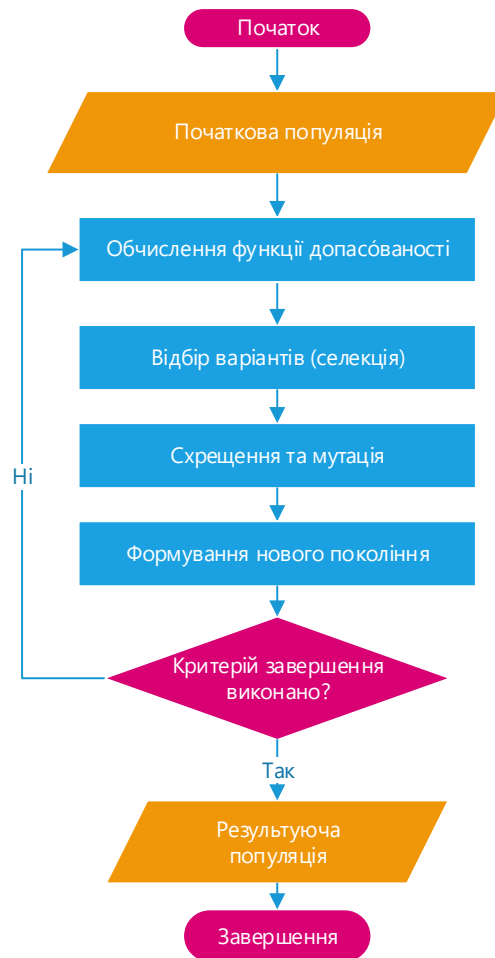


Рисунок 3.1 Граф-схема генетичного алгоритму

Крок 3: відбір варіантів з найліпшими функціями допасованості та виконаними угодами про рівень послуг (SLA).

Крок 4: схрещування – для отримання нових потомків вибрано 2-точкове схрещування

Крок 5: мутація – передбачає перебалансування, заміну, переміщення.

Крок 6: отримання нового покоління, перехід до нового етапу еволюції.

У нашому методі використовується 50 еволюцій. Отже, зрештою віртуальні машини будуть зіставлені з фізичними.

На першому кроці генетичного алгоритму виконується створення початкової популяції. Можливо застосування трьох різних евристичних стратегій ранжирування для перших трьох хромосом.

### Лістинг 3.1 – Псевдокод формування початкової популяції

```

For i from 0 to PopulationSize-2 do
  For j from 0 to ChromosomesSize-1 do
    produce at random a gene  $g_{ij}$ .
    If ( $Fs(g_{ij}) > Fs(g_{i+1j})$ ) Chromosomes $_{i+1}$  = Chromosomes $_i$ 
  End for
End for
Push Sub-Task to Priority Queue
While not Empty (Priority Queue)
  Population  $T_i$  (First Sub-Task) from Priority Queue
  For each processor  $P_k$  do
    Insertion-based HEFT Scheduling Policy()
    Assign  $T_i$  to the Processor  $P_k$ 
  End For
End While
Return Population  $T_i$ 

```

Розподіл обчислювальних завдань між процесорами виконується наступним чином. Вхідними даними виступають двоє батьків із існуючої популяції. Результатом є двоє нащадків. Алгоритм розподілу завдань містить наступні кроки.

Крок 1. Обирається точка перетину випадковим чином.

Крок 2. Хромосоми батька і матері розділяються на ліву і праву частини.

Крок 3. Створюється нове потомство, зокрема син. Лівий сегмент хромосоми батька переноситься на лівий сегмент хромосоми сина.

Крок 4. Гени з хромосоми матері, які не зустрічаються в лівому сегменті хромосоми батька, копіюються у правий сегмент хромосоми сина.

5. Створюється нове потомство, зокрема дочку. Лівий сегмент хромосоми матері переноситься на лівий сегмент хромосоми дочки.

6. Гени з хромосоми батька, яких немає в лівій хромосомі матері, переносяться у праву хромосому доньки.

7. Повернення (два обраних потомства).

Операція мутації представлено псевдокодом в лістингу 3.2.

### Лістинг 3.2 – Псевдокод мутації

Input: A chromosome  $T_i$  has been picked at Random.

Output: A new chromosome is being created.

Line 1:

```

from Succ(i) select the first successor Tj
In the interval [i+1, j-1] Choose a gene Tk randomly
For all Tl member Pred(k)
    If (l < i)
        Then Swapped(Ti, Tk) to create a new generation.
        Return the new offspring
    Else
        Go to Line 1
    End If
End for

```

Порядок відбору, який використовуються для виконання процесу мутації наступний:

Крок 1. Отримати результат алгоритму (лістинг 3.2).

Крок 2. Активувати менеджер завдань.

Крок 3. Почати обчислювальний процес.

Крок 4. Об'єднати результат в блоки розміром N.

Крок 5. Перевірити критерії завершення процесу.

Крок 6. Якщо всі члени популяції відповідають угодам про рівень якості послуг, передати отримане розподілення на виконання до кластеру.

### 3.3 Використання методу випадкового лісу

Випадковий ліс – це метод машинного навчання, який можна використовувати як для регресії, так і для класифікації. Це один з найбільш універсальних і простих для розуміння алгоритмів. Більше дерев означає більш стійкий випадковий ліс, і це справедливо для всіх випадкових лісів. У кожному вузлу випадковий ліс вибирає невелику кількість атрибутів для поділу, а потім обчислює оптимальний розподіл на основі цих характеристик,

як у навчальному наборі, роблячи кожне дерево унікальним. Коли все сказано і зроблено, він отримує прогноз із кожного дерева, а потім вибирає найкращий варіант шляхом «голосування більшістю» або «голосування за ефективність». На рисунку 3.2 показано граф-схему алгоритму застосування методу випадкового лісу щодо оптимізації процесу розподілення віртуальних машин та завдань.

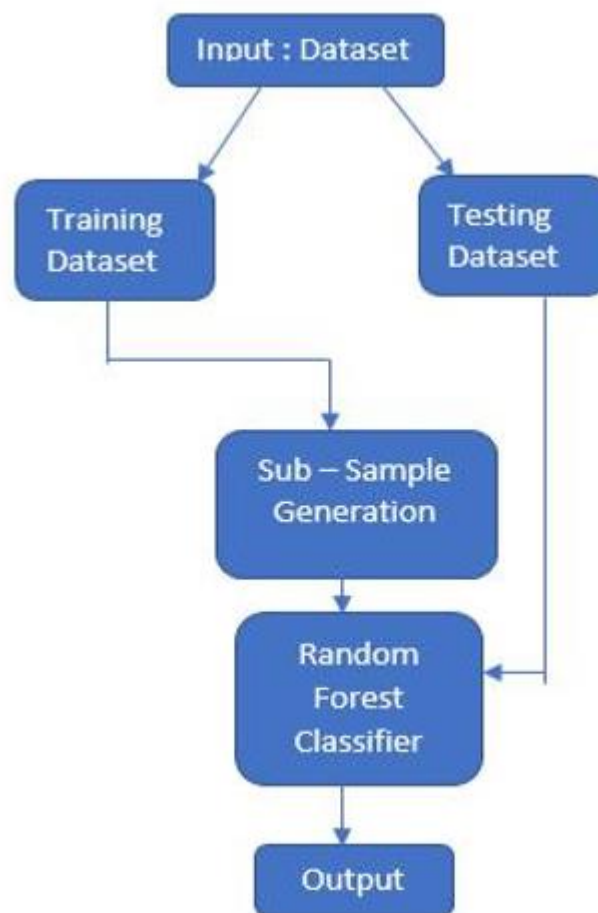


Рисунок 3.2 – Класифікація за допомогою методу випадкового лісу

Виконання починається після встановлення віртуальних машин на фізичних машинах. Через деякий час фізична машина може отримати більше віртуальних машин для виконання; у цьому випадку фізична машина стає перевантаженою. Для вирішення цього типу сценарію ми використали техніку міграції віртуальної машини.

Коли фізичні машини (PM) перевантажуються, деякі з віртуальних машин вибираються та переносяться на інші PM з меншим навантаженням, забезпечуючи баланс навантаження на всі фізичні машини в центрі обробки даних.

Одним із доступних методів виявлення перевантаження є інтерквартильний діапазон (IQR), який використовується для виявлення перевантажених фізичних машин у центрі обробки даних. Після цього віртуальну машину, яку потрібно перенести з перевантаженої фізичної машини, вибирають із використанням політики максимальної кореляції. Цей метод вибирає віртуальну машину з найвищим співвідношенням використання ЦП та інших віртуальних машин.

### 3.4 Оцінка ефективності запропонованих рішень

Першим критерієм ефективності є енергоспоживання. Фізичні машини (PM) у центрі обробки даних споживають енергію, і загальна спожита енергія дорівнює сумі усіх об'єктів споживання енергії, які обслуговують процес розподілених обчислень. Для розрахунку енергоспоживання фізичних машин використовувався лінійний кубічний метод. Модель живлення згідно цього методу припускає, що зі збільшенням використання центрального процесору, зростає і енергоспоживання фізичного хоста. Розглянемо наступні змінні для моделі потужності:

- хост  $k$  повністю використовується, максимальна споживана потужність становить  $E_k^{max}$  ;

- хост  $k$  знаходиться в режимі очікування, спожита потужність становить  $E_k^{idle}$  ;

- коефіцієнт використання процесорного модуля  $U_k^{CPU}$   $k$ -го вузла кластеру;

- $N$  – кількість вузлів кластеру;

- загальне споживання енергії:

$$E = \sum_{k=1}^N E_k = \sum_{k=1}^N ( E_k^{idle} + ( E_k^{max} - E_k^{idle} ) \times U_k^{CPU} ) . \quad (3.2)$$

Наступним важливим критерієм є час виконання завдань. З точки зору хмарного провайдера, це дуже важливо виконувати всі запити користувачів якомога швидше. Як результат, один із найважливіших показників ефективності для оцінки алгоритмів – це час виконання:

$$Time = \sum_{j=1}^M T_{VM_j}^{task} , \quad (3.3)$$

який оцінюється сумарним часом виконання усіх завдань усіма віртуальними машинами.

Хмарний центр обробки даних створює різні типи віртуальних машин для обробки запитів користувача на основі вимог користувача до ресурсів. Техніка VMP спрямована на покращення використання ресурсів шляхом розгортання віртуальної машини на відповідній фізичній машині. Враховано тип ресурсу ЦП. Залежно від кількості ресурсів, які запитує клієнт, хмарний центр обробки даних може створювати різні віртуальні машини (ВМ). Розміщення віртуальних машин на такій відповідній фізичній машині є однією з цілей підходу VMP. Сумарна продуктивність ЦП.

$$U_{cluster} = \frac{\sum_{k=1}^N U_k^{CPU}}{N} . \quad (3.4)$$

## 4 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

### 4.1 Опис середовища моделювання

Для перевірки запропонованого алгоритму ми використали симулятор набору інструментів CloudSim 4.0. Cloudsim пропонує різноманітні методи надання віртуальної машини та віртуалізовані ресурси. Для проведення експерименту використали відстеження реального робочого навантаження від PlanetLab. PlanetLab є компонентом проекту CoMon, який вимірює використання ЦП понад 1000 віртуальних машин, що працюють на різних хостах у понад 500 місцях по всьому світу. В експериментах використано чотири різні типи віртуальних машин: мікро, малі, середні та надвеликі екземпляри, кожна з яких має власну кількість MIPS. Розгорнуто 32 гетерогенних хоста.

Моделювання проводилося на 32 віртуальних машинах, які працювали на платформі Azure (фірми Microsoft), що розгорталися на фізичних машинах з процесорами, що містили 4 ядра кожна. Таблиця 4.1 задає конфігурацію кожної віртуальної машини у кластері.

Таблиця 3.1 – Апаратна конфігурація вузла (фізичної машини)

Хмарна платформа	Тип	Ядра	Процесор	Об'єм пам'яті	Частота	Базова прод. ЦП	Мережа
Azure	Standard_B4ms	4	Intel® Xeon® Platinum 8370C (Ice Lake)	32 GB	2,1Гц	22%	10 gigabit ethernet

## 4.2 Опис проведених експериментів з оцінки запропонованого методу

Кількість використаних хостів і віртуальних машин для моделювання коливалася від 4 до 32 та 32 до 256 відповідно залежно від конфігурації центру обробки даних. Набір даних PlanetLab є файлом журналу з центру обробки даних, який містить вхідний трафік, розмір завдання, розмір віртуальної машини. Віртуальна машина має такі конфігурації, як оперативна пам'ять, кількість процесорів тощо.

Наведені нижче показники використовуються для оцінки запропонованого алгоритму та інших алгоритмів. Вони представляють загальну кількість енергії, спожитої всіма фізичними машинами (PM) у центрі обробки даних. Лінійна кубічна модель енергоспоживання використовується для розрахунку енергоспоживання фізичних машин. У цій моделі живлення зі збільшенням використання центрального процесора споживання енергії фізичним хостом зростає лінійно. На рисунку 4.1 показано енергоспоживання активних фізичних машин. Random Forest проявив вище енергоспоживання за генетичний алгоритм на 13%, 38%, 49% і так далі.

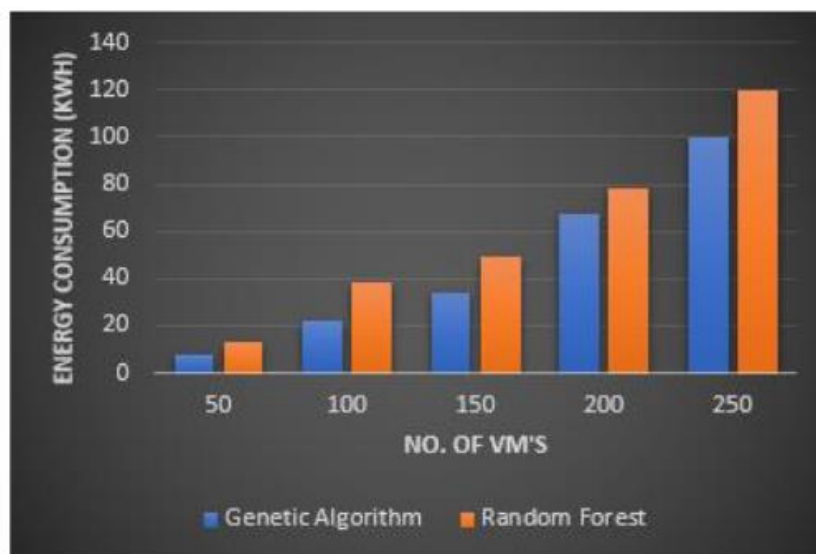


Рисунок 4.1 – Ефективність енергоспоживання генетичного алгоритму та алгоритму випадкового лісу

З точки зору хмарного постачальника, виконання всіх запитів користувачів за менший час є критичним. Як наслідок, одним із ключових факторів ефективності, який використовується для оцінки алгоритмів, є час виконання. На рисунку 4.2 показано час виконання активних фізичних машин. Випадковий ліс вище з точки зору генетичного алгоритму на 20%, 40%, 50% і так далі.

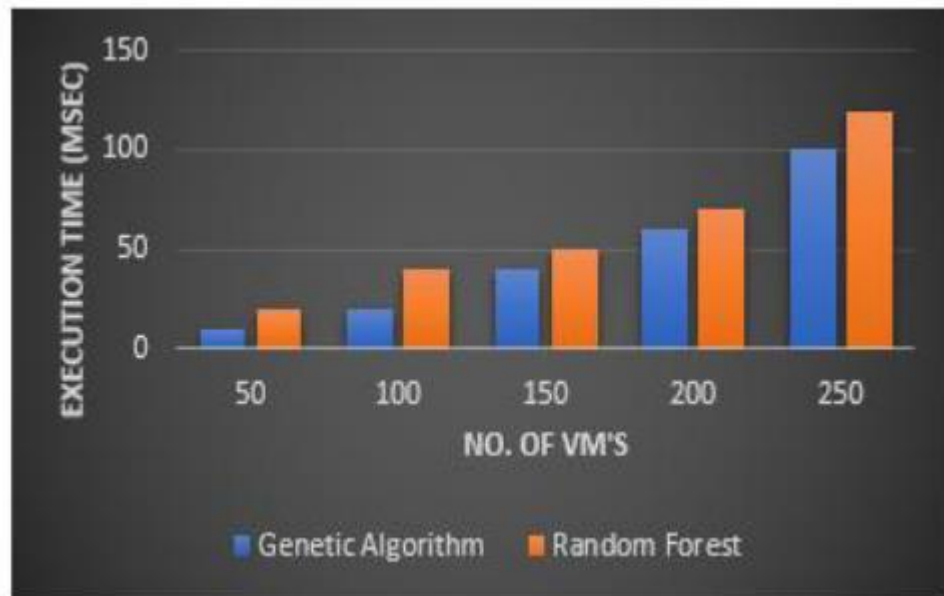


Рисунок 4.2 – Час виконання генетичного алгоритму та алгоритму випадкового лісу

На рисунку 4.3 показано використання ЦП активними РМ. Випадковий ліс показав більшу завантаженість процесорів ніж у генетичного алгоритму на 13%, 44%, 65% і так далі.

Часова складність є важливим фактором, який слід враховувати під час вивчення продуктивності алгоритму в хмарних обчисленнях. Існує кілька досліджень, які демонструють порівняльний аналіз складності генетичного алгоритму [23, 24]. Відповідно до дослідження, генетичний алгоритм знаходить найкраще глобальне оптимальне рішення за рахунок тривалого часу пошуку, але робить це краще, ніж алгоритми оптимізації колонії мурашок і оптимізації роїв частинок.

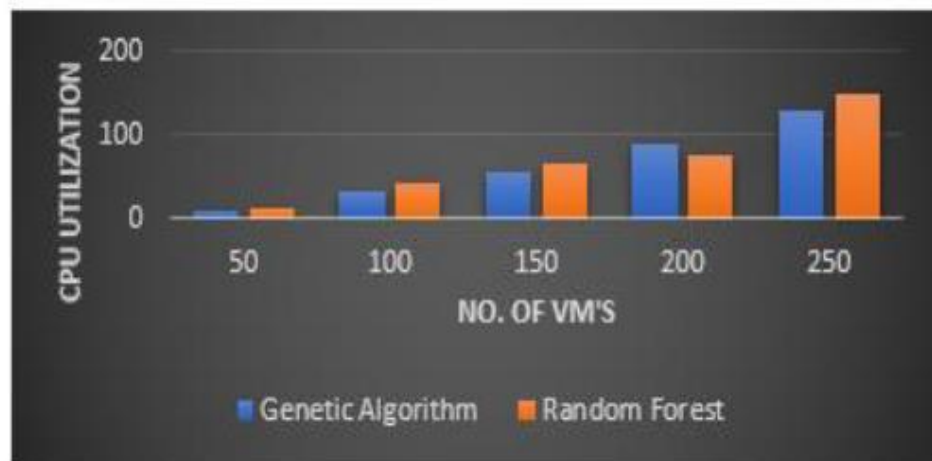


Рисунок 4.3 – Використання процесорного часу генетичним алгоритмом та алгоритмом випадкового лісу

Щоб скоротити час, необхідний для пошуку генетичного алгоритму, оптимальне рішення навчається на моделі випадкового лісу для навчання та подальшого прогнозування, що забезпечує оптимальне рішення в постійному часі. Накладні витрати виникають лише один раз під час створення оптимального рішення за допомогою генетичного алгоритму та навчання моделі.

#### 4.3 Опис проведених експериментів з порівнянням запропонованого методу з існуючими

Рисунок 4.4 показує порівняння споживання енергії для розробленого та існуючих методів. Кожен вузол має окремий процесор для обчислення, свою оперативну та зовнішню пам'ять. Це впливає на збільшення продуктивності обробки даних кластером. В усіх експериментах використовувалась та ж сама конфігурація мережевого обладнання для передачі даних. З рисунку видно, що енергоспоживання розробленого методу нижче за споживання енергії існуючими методами, а саме, на основі мурашиного алгоритму та оптимізації рою частинок (анг. particle swarm optimization, PSO). Загальне зниження енергоспоживання становить 7-8%.

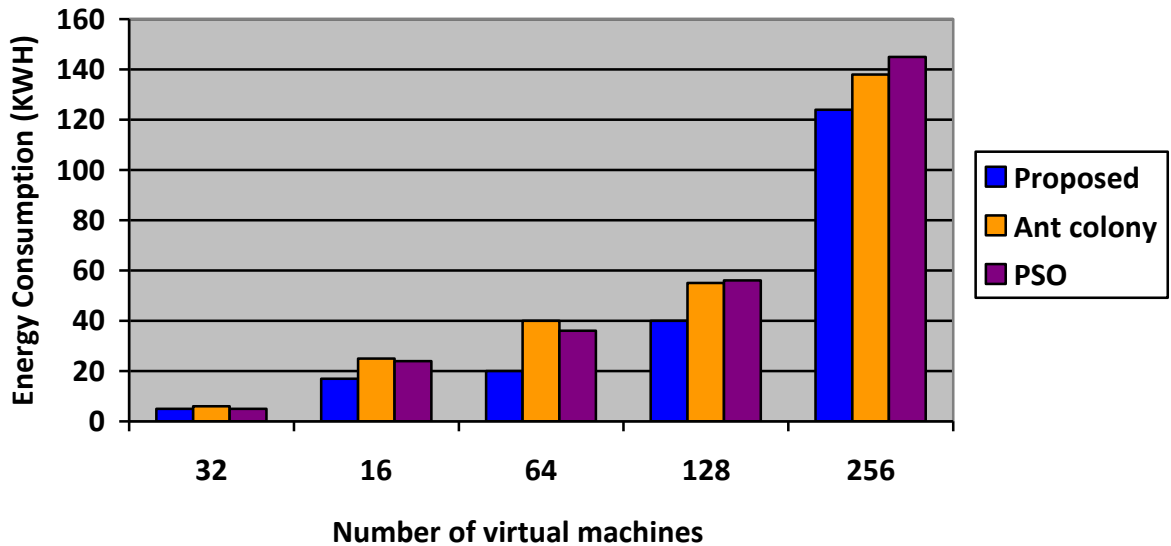


Рисунок 4.4 – Ефективність енергоспоживання розробленого методу в порівнянні з існуючими методами

Наступний експеримент показує залежність часу виконання пакету завдань, який містив 1000 програмних компонентів, між віртуальними машинами, що були призначені різними алгоритмами розподілу: мурашиним алгоритмом, алгоритмом оптимізації рою частинок та запропонованим алгоритмом. Аналіз графіку (рисунок 4.5) показує, що запропонований метод показує менший час виконання, чим при існуючі алгоритми. Середній показник зменшення часу на множині проведених експериментів складає 11-14%.

Якщо порівняти дані, попередніх двох експериментів, можна зробити висновок, що час обчислень зменшується більшими темпами ніж енергоспоживання. Даний факт можна пояснити наступним чином. Обчислювальна складність завдань залишається незмінною. Тому зменшення часу виконання забезпечується більш оптимальним використанням процесорів, які збільшують кількість обчислень на одиницю часу (коефіцієнт використання), що призводить до зростання енергоспоживання. Енергетична складова на передачу даних при цьому не змінюється.

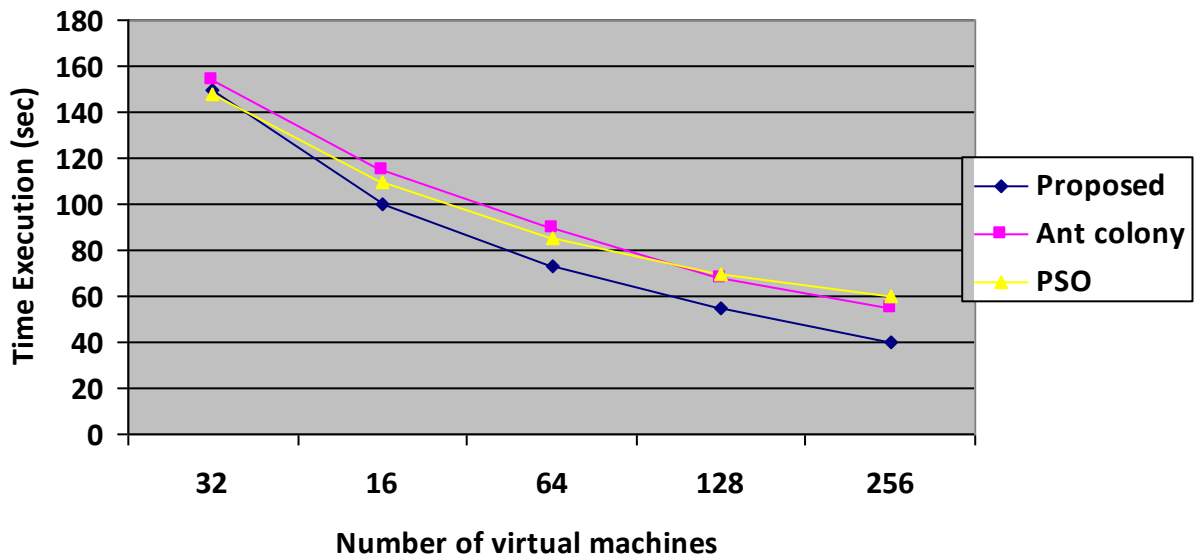


Рисунок 4.5 – Час виконання пакету завдань для розробленого методу в порівнянні з існуючими методами

#### 4.4 Виявлені проблеми з розподілу ресурсів

Під час роботи було виявлено декілька ситуацій під час розподілу ресурсів, що призводили до помилок в системі керування розподіленими обчисленнями та перезавантаженню програмного забезпечення:

- багато програм намагалися отримати доступ до одного ресурсу (процесору або каналу зв'язку) одночасно;
- дефіцит ресурсів, якщо ресурсів недостатньо для обслуговування пакетів завдань;
- використання ресурсів одночасно декількома програмами, до яких можуть відноситися програми з завдань (працюють на віртуальній машині) та локальні програми системи (працюють на фізичній машині паралельно з віртуальними машинами);
- надмірне надання ресурсів (коли програмі надається більше ресурсів, ніж їй потрібно), що призводить к зменшенню коефіцієнту використання ресурсів;

- зменшене надання (стан, який називається «менше надання», виникає, коли програма має менше ресурсів, ніж запитувала.

#### 4.5 Напрямки подальших досліджень

Сучасні хмарні платформи зробили можливим більш ефективний розподіл ресурсів. Для динамічного та ефективного розподілу ресурсів було розроблено багато методів планування. Оскільки споживання енергії безпосередньо впливає на доступність ресурсів і управління для забезпечення QoS у реалізаціях програм, необхідний належний моніторинг споживання енергії. Управління ресурсами та скорочення споживання енергії також є складним завданням для проектування через вимоги ефективного розподілу.

Обмеження щодо якості обслуговування (QoS) можна виконати за допомогою RAS-M. Енергозбереження також можна покращити, призначаючи віртуальні машини найбільш ефективним способом.

Порушення SLA можна уникнути, змінивши різні параметри SLA, і в майбутньому кількість параметрів SLA збільшиться для підвищення ефективності, а отже, буде відбуватися зменшення штрафів.

Енергозбереження: попередні моделі прогнозування зберігають електроенергію, вимикаючи вузли, які не використовуються. Однак це не забезпечує повної економії електроенергії. Щоб успішно економити електроенергію, можна передбачити додаткову обробку статистичних даних о системі, тому існуючі методи керування живленням можуть бути синхронізовані з методами планування. Системи управління хмарними очисленнями можуть отримати доступ до фізичного включення або виключення фізичних машин.

Ще одним напрямком досліджень може стати контроль заторів при передачі пакетів даних. Існуючі методи контролю заторів є кращими, але їх

можна покращити шляхом подальшого розвитку ідей, що були розглянуті в даній роботі.

Незважаючи на збереження необхідної якості обслуговування для програмних завдань та досягнення ефективності використання ресурсів, відкрите дослідження хмарних обчислень стикається зі складним завданням: виконати клієнтське програмне забезпечення за допомогою хмари. З огляду на майбутні цілі енергоефективності, планування та використання програмного забезпечення, як у хмарі, можна вивчати в майбутньому стратегію створення обчислювальних програм з урахуванням споживання енергії.

## ВИСНОВКИ

Як для компаній, так і для усього суспільства хмарні обчислення змінюють правила та технології життєвого циклу. Щоб досягти раціонального використання систем хмарних обчислень, необхідно звернути увагу на інновації даних. Зрештою, обробка в реальному часі замінює пакетне виконання завдань.

У контексті хмарних обчислень, стратегія визначення активів має важливе значення для задоволення вимог клієнтів і підвищення конкурентної переваги постачальників хмарних послуг. В роботі було кілька методів розподілу віртуальних машин за комп'ютерними ресурсами та пакетів програмних завдань за віртуальними машинами.

Крім того, у роботі детально досліджуються деякі проблеми розподілу ресурсів та їх вирішення. Однак це технологія, яка постійно розширює свої межі у відповідь на потреби та вимоги своїх користувачів. Розподіл ресурсів виступає як компонент хмари, який має найпоширеніше використання. На даний момент він покладається виключно на евристичні методи, тому що завдання повної багатокритеріальної оптимізації, на сьогодні, неможливо виконати.

ІТ-організації повинні застосовувати компоненти, пов'язані з умовами розподілу ресурсів користувачів, щоб бути ефективним підприємством, що швидко розвивається в ІТ-галузі. Цей підхід має потенціал для досягнення найкращих результатів щодо загальної продуктивності та задоволеності клієнтів. Як наслідок, розподіл ресурсів є складним і трудомістким завданням, яке потрібно виконати якнайшвидше, коли користувач запитує про це.

Лідери у галузі хмарних обчислень зможуть розподіляти ресурси між клієнтами, водночас одержуючи прибуток і задовольняючи їхні потреби за

різноманітних умов. Отже, наші результати показують кращі результати в ніж інші методи.

Використання двох методів: випадкового лісу та генетичного алгоритму забезпечує кращі результати порівняно з існуючими методами. Їх використанням можна досягти підвищення ефективності та швидкого вибору відповідного методу розподілу ресурсів. Тестування обраної моделі стане можливим у майбутньому за допомогою різних підходів до глибокого навчання для кращого вирішення проблем і аналізу продуктивності.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. P. Ray. A survey on internet of things architectures, *Journal of King Saud University-Computer and Information Sciences*. Vol.30 (3).2018. P.291–319. <https://doi.org/10.1016/j.jksuci.2016.10.003>
2. B. Safaei, A. Monazzah, M. Bafroei, A. Ejlali. Reliability side-effects in internet of things application layer protocols, in: *2nd International Conference on System Reliability and Safety (ICSRS)*.2017. P. 207–212. DOI: 10.1109/ICSRS.2017.8272822
3. S. Talari. A review of smart cities based on the internet of things concept, *Energies* 10 (4). 2017. P.1-23. <https://doi.org/10.3390/en10040421>
4. S. Noor. Energy demand side management within micro-grid networks enhanced by blockchain. *Applied Energy*. Vol. 228. 2018. P. 1385–1398. <https://doi.org/10.1016/j.apenergy.2018.07.012>
5. E. Borgia. The internet of things vision: key features, applications and open issues. *Computer Communication*. Vol. 54. P. 1–31. <https://doi.org/10.1016/j.comcom.2014.09.008>
6. A. Gupta, R. Christie, R. Manjula. Scalability on Internet of Things: Features, Techniques, and Research Challenges. *International Journal of Computational Intelligence Research*. Vol. 13. 2017. P. 1617-1627
7. A. Javed, Scalable iot platform for heterogeneous devices in smart environments, *IEEE Access* Vol 8. 2020. P. 211973–211985. DOI: 10.1109/ACCESS.2020.3039368
8. M. R. Palattella. Internet of things in the 5g era: enablers, architecture, and business models. *IEEE Journal on Selected Areas in Communications*. Vol.34, Issue 3. 2016. P. 510–527. DOI: 10.1109/JSAC.2016.2525418
9. T. Bakhshi. State of the Art and Recent Research Advances in Software Defined Networking. *Wireless Communications and Mobile Computing*. Article ID 7191647. 2017. P.1-35. <https://doi.org/10.1155/2017/7191647>

10. M. A. Razzaq. Simulation and assessment of vertical scaling for a smart campus environment using the internet of things, *IEEE Access* 10 96322–96330.
11. H. Z. Fahmi, F. J. Lin, Nfv-enabled vertical scalability for iot slices, in: *22nd Asia- Pacific Network Operations and Management Symposium*. IEEE access, vol. 10. pp. 96322-96330. 2022. doi: 10.1109/ACCESS.2022.3204042
12. M. Gusev. Scalable dew computing, *Appl. Sci.* Vol.12 (19) 9510. 2022. P.1-10. <https://doi.org/10.3390/app12199510>
13. W. Brockelsby, R. Dutta. Traffic analysis in support of hybrid sdn campus architectures for enhanced cybersecurity. *2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops*. Paris, France. 2021. P. 41-48. DOI: 10.1109/ICIN51074.2021.9385530.
14. S. Tang. Coordinate-based efficient indexing mechanism for intelligent iot systems in heterogeneous edge computing, *J. Parallel Distr. Comput.* Vol.166.2022. P. 45–56. <https://doi.org/10.1016/j.jpdc.2022.04.012>
15. W. Han, S. Li, The method of allocating resources for ideological and political education in universities based on iot technology. *Journal of Information & Knowledge Management* Vol. 21, No. Supp02, 2240011 (2022). <https://doi.org/10.1142/S0219649222400111>
16. W. Altwoyan, I. S. Alsukayti, A novel iot architecture for seamless iot integration into university systems, *Int. J. Adv. Comput. Sci. Appl.* 13 (4).
17. Y. Zheng, Q. Li, Y. Chen, X. Xie, W. Ma. Understanding mobility based on gps data. *10th International Conference on Ubiquitous Computing*. 2008 P.312–321. <https://doi.org/10.1145/1409635.1409677>
18. Y. Zheng, X. Xie, W. Ma. Geolife: a collaborative social networking service among user, location and trajectory. *IEEE Data Eng Bull.* Vol. 33(2). 2010. P. 32–39.
19. M. A. Vouk. Cloud computing : Issues, research and implementations. In *Information Technology Interfaces, 2018. ITI 2018. 30th International Conference on, , June 2018*. P. 31–40.
20. L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner. A break

in the clouds: Toward a cloud definition. *ACM SIGCOMM Computer Communication Review*, 39(1). 2019. P.50–55.

21. L. Johnson, A. Levine, and R. Smith. *The 2019 horizon report*, 2019. Austin, Texas, The New Media Consortium.

22. Foster, Y. Zhao, I. Raicu and S. Lu, “Cloud Computing and Grid Computing 360-Degree Compared,” *Grid Computing Environments Workshop (GCE '08)*, 2008.

23. R. Buyya, R. Ranjan and R. N. Calheiros, Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. *Proceedings of the Conference on High Performance Computing and Simulation*, Leipzig, Germany. IEEE Press: New York, U.S.A., 21–24 June 2019. pp.1–11.

24. L. Vaquero, L. Rodero-Merino , J. Caceres and M. Lindner, A break in the clouds: towards a cloud definition. *SIGCOMM Computer Communication Review*, Volume 39, Number 1, January 2019. pp.50-55.

25. NIST definition of cloud computing, [http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145\\_cloud-definition.pdf](http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf)

26. L. Youseff, M. Butrico and D. Da Silva, *Toward a Unified Ontology of Cloud Computing*, *Grid Computing Environments Workshop (GCE '08)*, 2008.

27. Six benefits of cloud computing. Available at <http://web2.sys-con.com/node/640237>.

28. B. Hayes. Cloud computing. *Communications of the ACM*, 51(7). July 2018. pp.9–11.

29. B. P. Rimal, E. Choi and I. Lumb, *A Taxonomy and Survey of Cloud Computing Systems*. *Fifth International Joint Conference on INC, IMS and IDC*. 2019, pp 44–51.

30. Gathering clouds of XaaS! [https://www.ibm.com/developerworks/mydeveloperworks/blogs/sbose/entry/gathering\\_clouds\\_of\\_xaas?lang=en](https://www.ibm.com/developerworks/mydeveloperworks/blogs/sbose/entry/gathering_clouds_of_xaas?lang=en)

31. B. Sonisky, chapitre 1, *cloud computing bible*, wiley publishing inc 2011

32. Q. Zhang, L. Cheng and R. Boutaba, Cloud computing : state of the art and research challenges. *Journal of Internet Services and Applications*, vol. 1, 2020 pp. 7–18.

33. C.N. Hofer and G. Karagiannis, Taxonomy of cloud computing services. In: *Proceedings of the 14th IEEE workshop on enabling the future service-oriented Internet (EFSOI'20), Workshop of IEEE GLOBECOM 2020*, pp. 1345–1350.

34. C. N. Hofer and G. Karagiannis, Cloud computing services: taxonomy and comparison, *Journal of Internet Services and Applications*, Springer, Vol. 2, September 2011, No. 2. pp. 69-79.

35. A. Lenk, M. Klems, J. Nimis, S. Tai and T. Sandholm. What's Inside the Cloud? An Architectural Map of the Cloud Landscape. In *ICSE Workshop on Software Engineering Challenges of Cloud Computing*, May 2019.

36. L. Wang and G. von Laszewski, Scientific Cloud Computing: Early Definition and Experience," in *20th IEEE International Conference on High Performance Computing and Communications, HPCCC'18*. 2-18. pp.825-830.

37. Q. Zhang, L. Cheng and R. Boutaba, Cloud computing : state of the art and research challenges. *Journal of Internet Services and Applications*, vol. 1. 2020. pp. 7–18.

38. Chiang, M. and Zhang, T., 2016. Fog and IoT: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6), pp.854-864.

39. Chen, M., Ma, Y., Song, J., Lai, C.F. and Hu, B.,. Smart clothing: Connecting human with clouds and big data for sustainable health monitoring. *Mobile Networks and Applications*, 21(5), 2016, pp.825-845.

40. Kumar, V., Laghari, A.A., Karim, S., Shakir, M. and Brohi, A.A., 2019. Comparison of Fog Computing & Cloud Computing. *International Journal of Mathematical Sciences and Computing (IJMSC)*, 5(1), pp.31-41.

41. Jayaraman, P.P., Perera, C., Georgakopoulos, D., Dustdar, S., Thakker, D. and Ranjan, R., 2017. Analytics-as-a-service in a multi-cloud environment through semantically-enabled hierarchical data processing. *Software: Practice and*

Experience, 47(8), pp.1139-1156.

42. Laghari, A.A., He, H., Karim, S., Shah, H.A. and Karn, N.K., 2017. Quality of experience assessment of video quality in social clouds. *Wireless Communications and Mobile Computing*, 2017.

43. Power, B. and Weinman, J., 2018. Revenue Growth is the Primary Benefit of the Cloud. *IEEE Cloud Computing*, 5(4), pp.89-94.

44. Benlian, A., Kettinger, W.J., Sunyaev, A., Winkler, T.J. and GUEST EDITORS, 2018. The transformative value of cloud computing: a decoupling, platformization, and recombination theoretical framework. *Journal of management information systems*, 35(3), pp.719-739.

45. Akherfi, K., Gerndt, M. and Harroud, H., 2018. Mobile cloud computing for computation offloading: Issues and challenges. *Applied computing and informatics*, 14(1), pp.1-16.

46. Mitra, A., O'Regan, N. and Sarpong, D., 2018. Cloud resource adaptation: A resource-based perspective on value creation for corporate growth. *Technological Forecasting and Social Change*, 130, pp.28-38.

47. Díaz, M., Martín, C. and Rubio, B., 2016. State-of-the-art, challenges, and open issues in the integration of the Internet of things and cloud computing. *Journal of Network and Computer Applications*, 67, pp.99-117.

48. Coppolino, L., D'Antonio, S., Mazzeo, G. and Romano, L., 2017. Cloud security: Emerging threats and current solutions. *Computers & Electrical Engineering*, 59, pp.126-140.

49. Marinescu, D.C., 2017. *Cloud computing: theory and practice*. Morgan Kaufmann.

50. Sultan, N., 2015. The implications of cloud computing for global enterprise management. *Global enterprise management* (pp. 39-56). Palgrave Macmillan, New York.

51. Galloway, S., 2017. *The four: the hidden DNA of Amazon, Apple, Facebook and Google*. Random House.

52. Gajbhiye, A. and Shrivastava, K.M.P., 2014, September. *Cloud*

computing: Need, enabling technology, architecture, advantages and challenges. In 2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence) (pp. 1-7). IEEE.

53. Aruna Irani, A.R., Manjula, D. and Sugumaran, V., 2019. Task scheduling techniques in cloud computing: A literature survey. *Future Generation Computer Systems*, 91, pp.407-415.

54. Verma, A., Pedrosa, L., Korupolu, M., Oppenheimer, D., Tune, E. and Wilkes, J., 2015, April. Large-scale cluster management at Google with Borg. In *Proceedings of the Tenth European Conference on Computer Systems* (p. 18). ACM.

55. Varghese, B. and Buyya, R., 2018. Next-generation cloud computing: New trends and research directions. *Future Generation Computer Systems*, 79, pp.849-861.

56. Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J. and Ghalsasi, A., 2011. Cloud computing—The business perspective. *Decision support systems*, 51(1), pp.176-189.

57. Laghari, A.A., He, H., Shafiq, M. and Khan, A., 2017, May. Impact of storage of mobile on quality of experience (QoE) at the user level accessing the cloud. In *2017 IEEE 9th international conference on communication software and networks (ICCSN)* pp. 1402-1409.

58. Aljamal, R., El-Mousa, A. and Jubair, F., 2018, April. A comparative review of high-performance computing major cloud service providers. In *2018 9th International Conference on Information and Communication Systems (ICICS)* (pp. 181-186). IEEE.

59. Belgaum, M.R., Soomro, S., Alansari, Z. and Alam, M., 2018. Cloud service ranking using checkpoint-based load balancing in real-time scheduling of cloud computing. In *Progress in Advanced Computing and Intelligent Engineering* (pp. 667-676). Springer, Singapore.

60. М.О. Волк, В.С Курочкін, А.П. Запорожченко, П.А. Паронікян. Гібридний метод розподілу ресурсів в хмарних системах. Системи управління, навігації та зв'язку. № 2 (76). 2024. С. 70-73. Фахове видання.