

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)

Метод організації туманних обчислень у динамічній
обчислювальній оверлейній мережі на базі полінових
мереж
(тема)

Виконав:

студент II курсу, групи СПМ-22-6
Гулько М.А.
(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: доц. Ткачов В.М.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

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 _____ Гуньку Михайлу Андрійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Метод організації туманих обчислень у динамічній обчислювальній
оверлейній мережі на базі полігрових мереж _____

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 _____

1) моделі полігрових мереж; _____

2) сучасні вимоги до туманих обчислень; _____

3) ОС Windows 10. _____

4. Перелік питань, що потрібно опрацювати у роботі _____

1) опис предметної області; _____

2) дослідження методу підвищення безпеки в полігрових мережах; _____

3) розробка методу організації туманих обчислень в полігрових мережах; _____

4) експериментальне моделювання; _____

5) висновки. _____

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

Слайд-презентація - 11 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	Огляд протоколів корпоративних комп'ютерних мереж	02.04.24-08.04.24	
2	Вибір та обґрунтування методики дослідження	09.04.24-16.04.24	
3	Вибір інструментальних засобів	17.04.24-22.04.24	
4	Розробка моделей протоколів	23.04.24-06.05.24	
5	Проведення експериментів	07.05.24-23.05.24	
6	Оформлення матеріалів кваліфікаційної роботи	24.05.24-03.06.24	
7	Подання кваліфікаційної роботи керівникові та її попередній захист	04.06.24-07.06.24	
8	Подання кваліфікаційної роботи на рецензування	08.06.24-12.06.24	

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Ткачов В.М.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 89 с., 16 рис., 19 табл., 2 дод., 19 джерел.

ПОЛІНГОВА МЕРЕЖА, ОВЕРЛЕЙНА МЕРЕЖА, ТУМАННІ ОБЧИСЛЕННЯ, ХМАРНІ ОБЧИСЛЕННЯ.

Метою кваліфікаційної роботи є розробка методу організації туманних обчислень у полігрових мережах.

У ході виконання кваліфікаційної роботи було детально проаналізовано особливості побудови систем з туманними обчисленнями, а також розглянуто специфіку використання полігрових мереж, зокрема сенсорних мереж. Вивчено різні підходи до інтеграції полігрових мереж у контексті сенсорних технологій, що дозволяє підвищити ефективність збору та передачі даних. На основі отриманих результатів було розроблено метод організації туманних обчислень, який враховує специфіку сучасних мережевих архітектур. Запропоновано нову архітектуру методу, яка сприяє покращенню продуктивності та надійності системи за рахунок оптимізації обчислювальних ресурсів та забезпечення більш ефективного управління даними. Також були проведені математичні експерименти для перевірки дієвості запропонованих рішень, що показали значні покращення в порівнянні з традиційними методами.

ABSTRACT

Master's thesis: 89 pages, 16 figures, 2 appendices, 19 sources.

POLING NETWORK, OVERLAY NETWORK, FOG COMPUTING,
CLOUD COMPUTING.

The major goal of this thesis is development of a method of organizing fuzzy calculations in polling networks.

In the course of the attestation work, the specifics of building systems with fog computing were analyzed in detail, and the specifics of using polling networks, in particular sensor networks, were also considered. Various approaches to the integration of polling networks in the context of sensor technologies have been studied, which allows to increase the efficiency of data collection and transmission. Based on the obtained results, a method of organizing fog computing was developed, which takes into account the specifics of modern network architectures. A new architecture of the method is proposed, which contributes to improving the performance and reliability of the system due to the optimization of computing resources and ensuring more efficient data management. Mathematical experiments were also conducted to verify the effectiveness of the proposed solutions, which showed significant improvements compared to traditional methods.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП	9
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Інтернет речей	11
1.2 Хмарні обчислення	14
1.3 Сервісні моделі	17
1.4 Моделі розгортання	18
1.5 Віртуалізація	19
1.6 Характеристики туманних обчислень для інтернету речей	24
1.7 Мобільні периферійні обчислення	26
1.8 Програмні фреймворки.....	28
1.9 Надання ресурсів.....	29
2 ДОСЛІДЖЕННЯ МЕТОДУ ПІДВИЩЕННЯ БЕЗПЕКИ В ПОЛІНГОВИХ МЕРЕЖАХ.....	32
2.1 Дослідження полігрових мереж	32
2.2 Дослідження оверлейних мереж.....	38
3 РОЗРОБКА МЕТОДУ ОРГАНІЗАЦІЇ ТУМАНИХ ОБЧИСЛЕНЬ В ПОЛІНГОВИХ МЕРЕЖАХ	45
3.1 Розробка архітектури моделі	45
3.2 Застосування оверлейних мереж	52
3.3 Визначення показників для перебудови ролей агентів	53
3.4 Розрахунок максимальної кількості вузлових агентів	54
4 МОДЕЛЬНИЙ ЕКСПЕРИМЕНТ	58
ВИСНОВКИ.....	64
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	66

ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	69
ДОДАТОК Б Фрагмент програмного коду.....	76

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- AI – штучний інтелект (англ., Artificial Intelligence)
- DTLS – протокол безпеки транспортного рівня для датаграм (англ., Datagram Transport Layer Security)
- IaaS – інфраструктура як послуга (англ., Infrastructure as a Service)
- IoT – Інтернет речей (англ., Internet of Things)
- ISDN – цифрова мережа інтегрованих послуг (англ., Integrated Services Digital Network)
- LAN – локальна мережа (англ., Local Area Network)
- M/G/1 – система масового обслуговування за класифікацією Кендалла (англ., M/G/1 queuing system)
- MAN – міська мережа (англ., Metropolitan Area Network)
- NFC – комунікація ближнього поля (англ., Near Field Communication)
- P2P – пірингові мережі (англ., Peer-to-Peer)
- PaaS – платформа як послуга (англ., Platform as a Service)
- QoS – якість обслуговування (англ., Quality of Service)
- RFID – радіочастотна ідентифікація (англ., Radio-Frequency Identification)
- RPL – протокол маршрутизації для мереж з низьким енергоспоживанням і високими втратами (англ., Routing Protocol for Low-Power and Lossy Networks)
- SaaS – програмне забезпечення як послуга (англ., Software as a Service)
- SLA – угоди про рівень обслуговування (англ., Service Level Agreement)
- VPN – віртуальна приватна мережа (англ., Virtual Private Network)
- VMM – монітор віртуальної машини (англ., Virtual Machine Monitor)
- VM – віртуальні машини (англ., Virtual Machines)
- WSN – бездротові сенсорні мережі (англ., Wireless Sensor Networks)

ВСТУП

У сучасному світі інформаційних технологій та обчислювальної науки, однією з найбільш перспективних галузей є розвиток та вдосконалення обчислювальних мереж. За останні десятиліття значний прогрес у цій області спостерігається завдяки постійному розширенню обчислювальних ресурсів, а також розвитку нових методів та алгоритмів обробки даних.

Однією з важливих гілок обчислювальної науки є туманні обчислення, що вивчаються як інструмент для моделювання та аналізу систем з нечіткими даними. Туманні обчислення знаходять своє застосування у багатьох сферах, включаючи штучний інтелект, розпізнавання образів, прийняття рішень та інші.

Однак, з огляду на зростання обсягів даних та складність обчислень, виникає потреба у розробці більш ефективних та масштабованих систем обробки інформації. Один із способів досягнення цієї мети – використання динамічних обчислювальних оверлейних мереж, які дозволяють ефективно використовувати розподілені ресурси для вирішення складних завдань.

Крім того, оверлейні мережі можуть бути використані для підвищення безпеки обчислювальних систем, забезпечуючи захищений канал передачі даних та ефективне управління доступом до ресурсів. Це є важливим аспектом, оскільки безпека даних і захист інформації є критично важливими у сучасних інформаційних системах.

Полінгові мережі є одним з ключових компонентів динамічних обчислювальних оверлейних мереж, які забезпечують ефективний обмін даними та координацію ресурсів між вузлами мережі. Використання полінгових мереж у поєднанні з туманними обчисленнями може дозволити покращити якість обробки даних та знизити час обчислень в розподілених системах.

Всі вищезгадані технології допоможуть розробити реконфігуровану модель туманних обчислень, яка б дозволила адаптуватися до змін у вимогах та умовах роботи системи. Така модель забезпечить гнучкість і стійкість обчислювальних процесів, підвищуючи ефективність використання ресурсів та зменшуючи затримки у передачі і обробці даних.

Таким чином, метою данної кваліфікаційної роботи є розробка методу туманних обчислень в оверлейних мережах на базі полінгових мереж. Також необхідно протестувати цей метод.

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Інтернет речей

IoT – це наразі процвітаюча технологічна тенденція, яку обговорюють у всьому світі. Термін «Інтернет речей» вперше було використано в презентації Procter & Gamble¹ в червні 1999[3]. У цій презентації Кевін Ештон поєднав радіочастотну ідентифікацію (RFID) із зароджуваним на той момент Інтернетом, що призвело до терміну Інтернет речей. Бачення IoT полягає в тому, щоб з'єднати величезну кількість розумних речей до Інтернету. Ці розумні речі можуть варіюватися від датчиків, приводів, носіїв, мобільних телефонів через одноплатні комп'ютери до мікроконтролерів. Нижче наведено ці речі називають пристроями IoT. Щоб підключити згадані пристрої IoT до Інтернету багато використовуються такі технології, як WiFi, Bluetooth, RFID і Near Field Communication (NFC). Дві останні згадані технології, RFID і NFC, використовуються близько близькості до пристроїв IoT для їх ідентифікації, відстеження або автентифікації.

Озираючись назад, можна зрозуміти, що RFID була першою технологією, яка привернула увагу IoT. Навіть сьогодні RFID є однією з найбільш рушійних сил щодо оцінки додатків IoT. Це стосується загального бачення, яке об'єднує все до Інтернету можна прямо застосувати за допомогою так званих тегів RFID. RFID-мітки є невеликі однозначно ідентифіковані та програмовані мікропроцесори, підключені до антени для зв'язку з пристроями зчитування RFID. Додаткові компоненти, наприклад, акумулятори, датчики GPS можуть бути додані в залежності від специфікацій застосування. Теги можна розмістити на будь-якому уявному предметі, наприклад, упаковці харчових продуктів, продуктах, які можна транспортувати, і навіть тіла тварин чи людей. Позначення об'єктів дозволяє ідентифікувати, відстежувати або засвідчити їх. Прикладами застосування тегів RFID є бібліотечні системи,

логістика відстеження товару та системи контролю доступу. З огляду на величезне покриття RFID-міток на повсякденних речах, кількість необхідних тегів є дуже великою. З величезною кількістю тегів потрібно ціна одиниці стає важливою. Отже, падіння ціни на RFID-мітки позитивно впливає на цю сферу, підвищуючи інтерес до програм за допомогою RFID.

Окрім RFID, цікавим є підхід бездротових сенсорних мереж (WSN). технологія щодо майбутньої тенденції IoT. Хоча WSN і IoT здається, що це конкуруючі технології, існує багато синергії між ними. Сенсорні мережі в WSN складається з бездротових взаємопов'язаних датчиків, здатних обмінюватися даними. Підхід спільного спілкування, який використовується в WSN, – це технологія Peer-to-Peer (P2P). Треба виявляти інші датчики в безпосередній близькості та будувати мережу для зв'язку на високому рівні георозподіленої сенсорної сітки. Прикладами таких WSN є датчики тепла та диму при пожежі лісів. У разі пожежі чи диму датчики надсилають екстрений виклик на проміжне програмне забезпечення для повторної перевірки надзвичайної ситуації або безпосередньо до пожежної частини.

Ще одна нова технологічна концепція, яка зараз обговорюється майже в кожній компанії з виробничим досвідом є Індустрія 4.0[5]. IoT тісно пов'язаний із цією концепцією Індустрії 4.0. Індустрія 4.0 – це бачення процесу цифровізації та автоматизації виробництва, починаючи з першого проекту продукту через повний ланцюг постачання до кінцевого продукту. Це включає в себе галузі аналітики великих даних, цифровізації, виробництва, штучний інтелект тощо.

У поточній ситуації IoT складається не лише з датчиків і приводів, але й як згадувалося раніше, з багатьох різномірних пристроїв, підключених до Інтернету. Це підкреслює необхідність стандартизованої техніки зв'язку між IoT пристроями для контролю, моніторингу та вилучення накопиченої інформації. Крім того, необхідно враховувати енергоефективність та обчислювальні ресурси. Ці рюми вимоги та можливості застосування залучають академії та індустрії інвестувати в дослідження IoT. Тому багато

відомих компаній, таких як Gartner Inc., Siemens AG проводять дослідження, щоб реалізувати власне бачення IoT.

Рушійні сили мають вирішальне значення для нових технологій. У випадку IoT – найбільш важливими рушійними силами є зменшення вартості потужності обробки та зберігання поширення даних, еластичний аналіз великих даних у хмарі та конвергенція цифрового та оперативного промислового світу. Крім того, один із більш очевидною рушійною силою є зручність використання, отримана завдяки новим можливим додаткам, наприклад, розумні міста, розумні будинки, розумні автомобілі, розумні мережі, розумна охорона здоров'я, розумна логістика, автономні автомобілі та роботи, а також програми віртуальної та доповненої реальності. Щодо розумних будинків, функціональність керування всім будинком за допомогою смартфона, планшета або носимого пристрою, включаючи можливість закривати вікна, змінювати температуру, контролювати яких продуктів недостатньо, або спостереження за літніми людьми, відповідає вимогам і діяльності майбутнього.

У прогнозі, зробленому Cisco Systems Inc., ступінь поширення пристроїв IoT з початку 2000-х до 2020 продемонстровано неймовірно зростання розумних об'єктів, тобто пристроїв Інтернету речей[7], з 6,3 мільярдів у 2003 році до приблизно 50 мільярдів у 2020 році. надзвичайний розвиток цього сектора.

Застосування, що з'являються в рамках цієї постійно зростаючої тенденції, поширюються на різні сфери. У багатьох прикладних сценаріях, пов'язаних із IoT, з'являється слово smart, що вказує на те самоадаптивний підхід. Самоадаптивна система динамічно переналаштовується відповідно до змін у навколишньому середовищі. Більш детально, самоадаптація об'єднує теми самовідновлення, самоуправління, самоконфігурації та самооптимізації.

Підсумовуючи, IoT є дуже багатообіцяючою новою технологією, яка набуває все більшого поширення інтерес і дослідницька увага в багатьох різноманітних областях. Можливі застосування і отримані переваги роблять

цю технологію дуже потужною. Нарешті, бачення та розвиток технології, які надходять із IoT, впливають не лише на конкретні продукти чи галузі, але здатні суттєво змінити погляд і вимоги до систем і додатків.

1.2 Хмарні обчислення

Хмарні обчислення – це широко досліджена централізована обчислювальна парадигма, яка робить наголос на динамічному забезпеченні обчислювальних ресурсів і ресурсів зберігання. Ці ресурси знаходяться в централізованих дата центрів. Вибір місця розташування дата центру пов'язано з багатьма факторами, наприклад, вартістю навколишньої енергії, температурою та цінами на землю. Ресурси, надані хмарними провайдерами, включають програмні послуги, що використовуються через Інтернет веб-переглядач, платформи розробників для створення та розгортання хмарних додатків і завершити серверні інфраструктури, що обробляють віртуальні машини (VM), що працюють на хмарних ресурсах. Ці три сервісні моделі доставки хмарних ресурсів називаються програмним забезпеченням як послугою (SaaS), платформа як послуга (PaaS) та інфраструктура як послуга (IaaS). Поруч з цими різними моделі обслуговування, з'явилися чотири різні моделі розгортання хмари: (i) приватна хмара, (ii) публічна хмара спільноти, (iii) публічна хмара та (iv) гібридна хмара. Деталі цих моделей розгортання представлені нижче. Націлюючись на динамічне надання величезних обсягів ресурсів на вимогу на основі оплати за використання, хмарні провайдери використовують технологію віртуалізації.

Основні характеристики хмарних обчислень були представлені у визначенні NIST у 2011 р. і досі охоплюють найважливіші його аспекти. Далі описані згадані характеристики.

Надання на вимогу. Хмарні послуги можна отримати на вимогу, і вони надаватимуться без участі людини, оскільки послуги та ресурси надаються автономно.

Вільний доступ до мережі. Оскільки хмарні обчислення дата центрів поширені по всьому світу, кожен пристрій, підключений до Інтернету, може взаємодіяти з хмарою та отримувати запропоновані послуги. Пристрої включають мобільні пристрої, переносні пристрої та пропріетарні сервери.

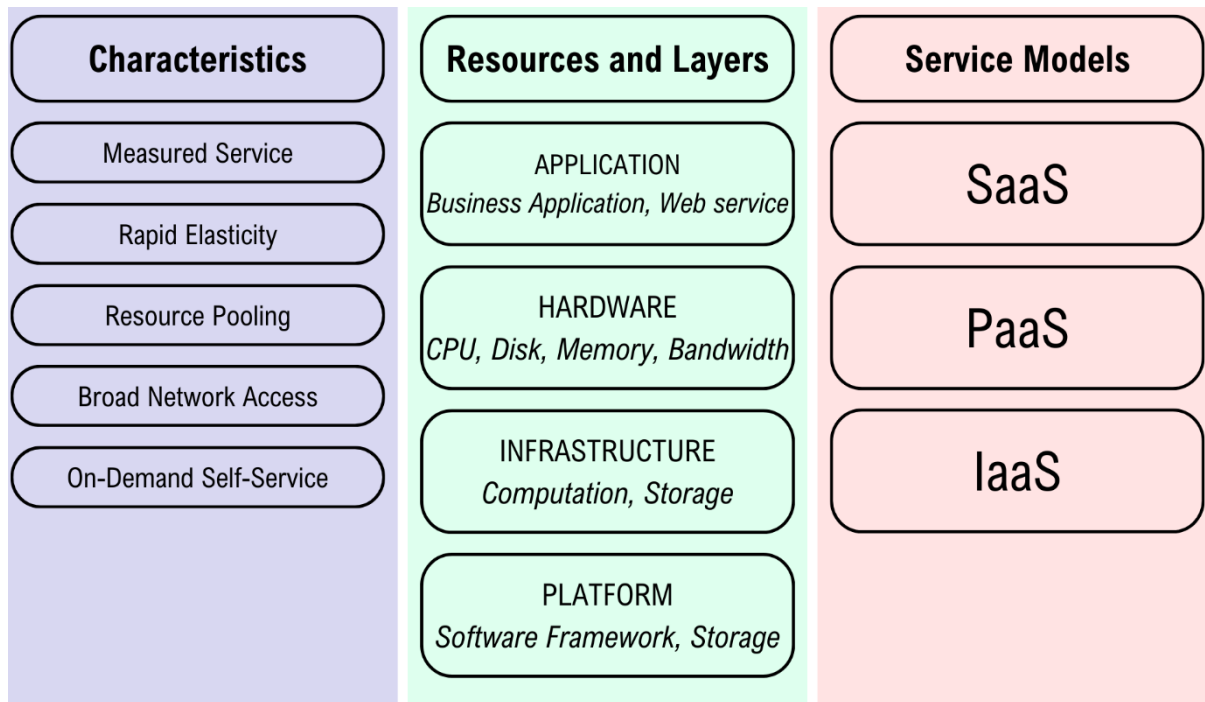


Рисунок 1.1 – Огляд хмарних обчислень

Зараз багато розумних пристроїв підключаються до Інтернету за допомогою хмарних сервісів. Це так, оскільки багато постачальників послуг уже використовують централізовані хмарні ресурси, наприклад Netflix, Airbnb, Slack, щоб отримати вигоду від динамічної обробки ресурсів і практично нескінченної ємності ресурсів.

Клієнти хмарних обчислень з об'єднанням ресурсів хочуть отримати динамічні ресурси відповідно до поточного попиту, щоб мінімізувати надлишок і недостатню кількість ресурсів. Надлишок ресурсів відбувається, коли виділена потужність ресурсів перевищує поточний попит, отже, ресурси витрачаються даремно. З іншого боку, недостатнє забезпечення – це сценарій, коли доступних ресурсів занадто мало для обслуговування попиту клієнтів.

Очевидно, що хочеться уникнути обох ситуацій і забезпечити оптимальну кількість ресурсів.

Завдяки концепції об'єднання ресурсів можна використовувати фізичні ресурси відповідно до попиту клієнтів. Традиційні дата центри мають статичні ресурси, оскільки фірмові сервери не можуть динамічно налаштовуватися без зміни інфраструктури.

З метою створення еластичного ресурсного середовища хмара складається з пулу ресурсів, тобто серверів із обробкою та ресурсами зберігання, які можна налаштувати відповідно до вимог клієнтів. Завдяки віртуалізації ресурсів можна легко вказати фізичні ресурси та керувати ними. Таким чином, фізичні сервери використовуються одночасно декількома клієнтами, тобто в режимі кількох орендарів. Метою наявності віртуального пулу ресурсів є обфускація різнорідних аспектів, щоб полегшити використання та композицію узагальнених ресурсів.

Швидка масштабованість. Дуже важливою характеристикою парадигми хмарних обчислень є швидка масштабованість хмари. Оскільки пул фізичних ресурсів використовується за допомогою технології віртуалізації ресурсів, ресурси можна масштабувати горизонтально та вертикально з мінімальними зусиллями щодо налаштування та керування. Горизонтальне масштабування – це розгортання додаткових екземплярів ВМ для одночасного обслуговування більшої кількості клієнтів, тоді як вертикальне масштабування – це адаптація конкретних можливостей існуючих ВМ. Завдяки цим можливостям динамічного масштабування попит клієнтів можна обробляти ефективніше. Таким чином можна зменшити енергоспоживання, витрати ресурсів і загальну вартість.

Контрольоване обслуговування. Оскільки хмарні послуги використовуються на основі оплати за використання, повне спілкування та взаємодія вимірюються та контролюються. Це не тільки для цілей виставлення рахунків, але й дає змогу хмарним постачальникам і клієнтам вимірювати якість обслуговування (QoS). Перш ніж придбати хмарні ресурси, замовник і

хмарний постачальник погоджуються на основі конкретних показників якості послуги, тобто угод про рівень обслуговування (SLA). Якщо ці угоди порушуються, провайдеру хмарних технологій, можливо, доведеться заздалегідь сплатити штраф, визначений у SLA.

1.3 Сервісні моделі

Сервісні моделі описують, як можна отримати та використовувати хмарні ресурси, розташовані в хмарному середовищі. Розташування та право власності на ресурси в цих диференціаціях не враховуються.

SaaS. Перша модель називається SaaS і є найбільш обмеженою можливістю. Клієнти SaaS отримують доступ до служби через веб-інтерфейс за допомогою веб-браузера або інтерфейсу програмування. Клієнт не може керувати чи спеціально налаштовувати фізичні хмарні ресурси служби. Прикладами SaaS є Microsoft Office 365² і Google Apps³.

PaaS. Більш гнучкою моделлю, спеціалізованою для розробників, є PaaS. У цій моделі надається структура розробки, яка дозволяє розробникам розробляти, тестувати та розгортати програми в хмарному середовищі. Націлена на створення гнучкого програмного середовища, хмара займається управлінням і контролем необхідних фізичних ресурсів. На додаток до загального управління та контролю ресурсів, хмара обробляє динамічне масштабування, щоб гарантувати, що узгоджені SLA не порушуються в будь-який момент часу. Прикладами PaaS є Google App Engine⁴ і Windows Azure⁵.

IaaS. Останньою та найбільш гнучкою моделлю послуг є інфраструктура як послуга. Тут клієнт має повний контроль над віртуальною машиною, що працює на фізичних хмарних ресурсах. Віртуальну машину можна налаштувати відповідно до необхідних характеристик, таких як оперативна пам'ять, процесор, сховище та мережеві функції. На цій віртуальній машині користувач може розгортати будь-які програми на вибраній операційній системі. Тим не менш, користувач не має повного

контролю над фізичними ресурсами, а лише над придбаною віртуальною машиною. Прикладами IaaS є Amazon Web Services⁶ і Open Stack⁷.

1.4 Моделі розгортання

Приватна хмара. Приватна хмара – це підхід, коли організація використовує хмарне середовище, явно призначене для цієї організації. Володіння, управління та експлуатація хмарних ресурсів може здійснюватися самою організацією, сторонньою організацією або поєднанням обох. Перевага цієї моделі полягає в тому, що дані, надіслані до приватної хмари, зберігаються лише в середовищі, яке використовується для цієї конкретної організації, тобто конфіденційні дані не передаються жодній іншій організації, пов'язаній із середовищем. З іншого боку, якщо ця конкретна приватна хмара виходить з ладу, дані можуть бути заблоковані в цій єдиній точці збою.

Обмежена(внутрішня) хмара. Обмежена хмара базується на тій самій ідеї, що й приватна хмара, але хмарне середовище призначене для використання в рамках певної спільноти. Крім того, власність, управління та експлуатація можуть здійснюватися однією або кількома організаціями, що утворюють громаду.

Публічна хмара. У загальнодоступній хмарі хмарне середовище не призначене для певної організації чи спільноти. Отже, він може використовуватися широким загалом. Власність, управління та експлуатація можуть здійснюватися урядом, академічними організаціями, будь-якою іншою організацією або їх комбінацією. Перевагою публічної хмари є відкрита доступність для всіх без будь-яких обмежень для конкретних організацій. Таким чином, негативним впливом відкритої доступності є безпека даних і послуг.

Гібридна хмара. Гібридна хмара – це комбінація приватної та публічної хмари. Таким чином, деякими хмарними ресурсами можна володіти, керувати та керувати ними в приватній хмарі, тоді як інші використовуються через

відкриту доступну публічну хмару. Наприклад, конфіденційні дані можуть зберігатися в приватній хмарі, а більш загальні некритичні дані – у публічній хмарі.

1.5 Віртуалізація

Віртуалізація описує абстракцію фізичних апаратних ресурсів комп'ютерної системи. Концепція віртуалізації є однією з основних рушійних сил хмарних обчислень з наступних двох причин. По-перше, фізичне обладнання може використовуватися кількома клієнтами ізольовано. Таким чином, переваги полягають у тому, що одна фізична апаратна система може служити багатьом клієнтам як основа для їх обробки, зберігання та запитів пам'яті, використовуючи більшість апаратних ресурсів машин.

Існують різні концепції віртуалізації, але в ході цієї дипломної роботи розглядаються лише повна віртуалізація та віртуалізація операційної системи. Остання є віртуалізацією на базі контейнера.

Повна віртуалізація. Абстракція повної віртуалізації призводить до створення віртуальної копії фізичних ресурсів, упакованих у віртуальну машину. Перед розгортанням віртуальної машини користувачеві потрібно вказати кількість процесорів, оперативної пам'яті та інших можливостей, якими має володіти віртуальна машина. Очевидно, кількість віртуальних машин, створених на одній машині, обмежена її фізичними ресурсами. Оскільки віртуальні машини, що працюють на фізичних ресурсах, повністю ізольовані за допомогою координаційного проміжного програмного забезпечення, тобто монітора віртуальної машини (VMM), системи залишаються повністю розділеними. По-друге, різноманітні апаратні компоненти можна об'єднати в однорідний пул ресурсів. Цей механізм об'єднання ресурсів використовується, щоб запропонувати різноманітні параметри віртуальної машини, наприклад, ЦП, ОЗУ та ємності для зберігання.

Повна віртуалізація апаратних ресурсів, тобто повна віртуалізація, працює наступним чином. З метою ізольованого середовища апаратне забезпечення керується згаданим VMM. VMM відповідає за моніторинг, керування та розгортання віртуальних машин поверх нього, запобігаючи прямому апаратному доступу до фізичного обладнання. Кожна віртуальна машина складається з окремої операційної системи, тобто гостьової операційної системи, і довільних програм, що працюють на ній (рисунок 1.3(a)). Переваги цього способу віртуалізації фізичних апаратних ресурсів полягають у гнучкості розгортання, простому забезпеченні адаптованих ресурсів і можливості компонувати гетерогенне обладнання. Негативними сторонами цієї конкретної концепції віртуалізації є тривалий час запуску та великий обсяг необхідного місця для зберігання. Обидва ці негативні аспекти викликані необхідністю окремої операційної системи в кожній віртуальній машині. Відомим підприємством, що розробляє рішення віртуалізації, є VMWare8.

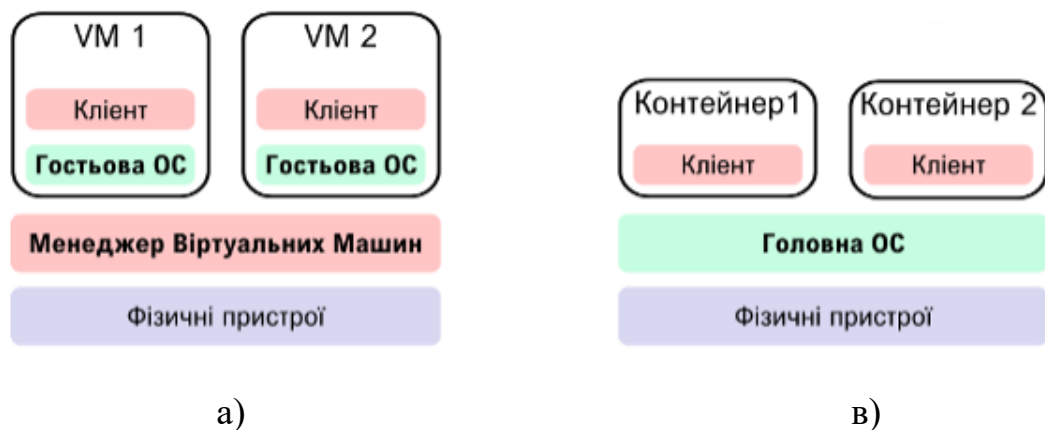


Рисунок 1.3 – Порівняння віртуалізації: а) повна віртуалізація; в) контейнерна віртуалізація

Віртуалізація операційної системи (контейнер). На відміну від згаданої раніше концепції віртуалізації, віртуалізація операційної системи або віртуалізація контейнера є дуже легкою концепцією віртуалізації. Концепція віртуалізації контейнерів побудована на основі існуючої операційної системи

хоста. Логічно відокремлені віртуальні середовища, тобто контейнери, працюють поверх однієї операційної системи хоста та використовують те саме ядро та загальне фізичне обладнання. Хоча ці контейнери працюють в одній операційній системі, віртуальне середовище повністю ізольоване. Іншими словами, контейнер може просто отримати доступ до просторів зберігання та процесів, пов'язаних із його власним контейнером.

Одним із найбільш використовуваних контейнерних технологічних рішень є Docker⁹. Docker розрізняє Docker Images і Docker Containers. Зображення Docker складається з кількох шарів, які зберігають знімок контейнера Docker. Потім Docker використовує файлову систему об'єднання, щоб об'єднати ці шари в одне зображення. Цей образ може бути створений за допомогою середовища виконання Docker, і результатом буде розгорнутий контейнер Docker. Контейнер Docker складається з базового образу (полегшеної операційної системи), доданих користувачем файлів і метаданих. Цей контейнер Docker можна запускати, зупиняти або зберігати знімок, тобто зображення Docker, і використовувати для розгортання ідентичних контейнерів. Додаткові інструменти оркестровки Docker для покращення можливостей простих контейнерів Docker включають, наприклад, Docker Machine, Docker Swarm і Docker Compose¹⁰. Docker Machine допомагає розгорнути механізми Docker на локальній машині, хмарних провайдерах або інших центрах домену. Docker Swarm використовується для кластеризації контейнерів Docker, а Docker Compose допомагає запускати розподілену систему, що складається з кількох контейнерів Docker.

Перевага контейнерів порівняно з віртуальними машинами полягає в тому, що контейнери можна запускати миттєво, оскільки їм не потрібна окрема операційна система для кожного контейнера. Контейнери мають невелику вагу і їх можна розгорнути, випустити та оновлювати дуже швидко, оскільки через спеціальну файлову систему об'єднання потрібно оновлювати лише певні шари. Отже, у випадку фреймворку туманного обчислення контейнери краще відповідають потребам.

Docker часто використовується в поєднанні з наразі процвітаючою архітектурою мікросервісів (MSA). MSA – це новий підхід до розробки невеликих, легких, незалежних, розподілених програмних компонентів. Один окремих програмний компонент називається *Microservice* і розгортається в незалежному контейнері. Деякі з найважливіших переваг мікросервісів: неоднорідність технології, стійкість, масштабування, легкість розгортання та оптимізація для заміни. Оскільки мікросервіси спілкуються за допомогою стандартизованих форматів, таких як JavaScript Object Notation (JSON) і Extensible Markup Language (XML), ці технології є абсолютно незалежними. Стандартною комунікаційною технологією, яка використовується в MSA, є Representational State Transfer (REST).

В останні кілька десятиліть обчислювальні моделі змінювали централізований і децентралізований підходи до обчислень. Починаючи з мейнфреймів у 70-х і 80-х роках, за еволюцією моделі послідувала хвиля децентралізації клієнт-серверної моделі в 90-х роках. Ця перша хвиля була спровокована падінням цін на персональні комп'ютери та зростанням інтересу до володіння власною обчислювальною потужністю. На початку 2000-х років обчислювальна модель знову перейшла від децентралізованого до централізованого підходу, а саме до парадигми хмарних обчислень. Незважаючи на те, що хмарні обчислення процвітають і не будуть замінені в найближчому майбутньому, існує сила, яка просуває новий децентралізований підхід до вирішення постійних проблем централізованих систем, наприклад, висока затримка, відсутність інформації про місцезнаходження. Різниця порівняно з попередньою парадигмою полягає в тому, що ця наступна парадигма не замінить попередню, а розширить її, щоб покращити певні можливості. Цей поточний перехід від парадигми централізованих хмарних обчислень до парадигми децентралізованих обчислень ознаменував народження туманних обчислень.

Туманні обчислення були представлені як нова технологія для IoT Бономі та ін. у 2012 році. Ця нова парадигма розподілених обчислень містить

ідею надання обчислювальних можливостей і можливостей зберігання ближче до пристроїв IoT, що створюють дані, на межі мережі. З метою зменшення відстані між кінцевими пристроями та найближчим блоком обробки ця парадигма вводить додатковий рівень багатих ресурсами пристроїв IoT, тобто туманних осередків. Ці туманні комірки мають власні обчислювальні можливості та можливості зберігання для обробки запитів завдань, фільтрації та попередньої обробки даних. Це створює відстань в один стрибок до кінцевих пристроїв і, отже, зменшує затримку та час виконання завдання. Іншим важливим розширенням обчислень у тумані є широке географічне поширення цих додаткових пристроїв, спрямованих на безперебійне та надійне виконання послуг навіть при підключенні до рухомих пристроїв, наприклад, інтелектуальних автомобілів, мобільних телефонів.

Більш детально, комірки туману розташовані на краю мережі, щоб зменшити затримку, час виконання завдання та кількість даних, що надсилаються через мережу. Потім осередки туману отримують запити на завдання від підключених пристроїв IoT, наприклад, датчиків і приводів або зовнішніх ініціаторів, і вирішують, де обробляти ці запити. Залежно від можливостей туманних осередків, компонент міркування вирішує, чи запит завдання обробляється локально чи поширюється в хмару.

Спілкування всередині обчислювального середовища туману, тобто ландшафту туману, не обмежується будь-якою формою зв'язку і тому може відбуватися через WiFi, стільникові мережі, Bluetooth або Ethernet. Щоб отримати надлишкове середовище, пейзаж туману також можна підключити до кількох хмарних постачальників або використовувати приватні хмари всередині ландшафту туману.

На рисунку 1.2 зображено графічне зображення туманного ландшафту. У нижній частині архітектури розташовані різноманітні пристрої IoT, за якими йдуть пристрої маршрутизації, підключені до осередків туману. Ці пристрої маршрутизації символізують можливість повторного використання існуючих мережевих пристроїв, наприклад, маршрутизаторів. Крім того, осередки

туману можуть бути підключені до інших осередків туману, або до загальнодоступних чи приватних хмар.

1.6 Характеристики туманних обчислень для інтернету речей

Нижче наведені основні характеристики парадигми туманних обчислень, які допомагають IoT використовувати свій потенціал.

Низька затримка та визначення місцезнаходження. Оскільки туманні комірки, розташовані на краю мережі, зменшують відстань між пристроями IoT і хмарними ресурсами, затримку та час виконання завдань можна значно скоротити. Крім того, через розташування поінформованості про туманні осередки можуть бути надані послуги позиціонування, наприклад, кешування залежного від розташування контенту. Ці послуги здатні попередньо обробляти, фільтрувати або кешувати запитаний контент.

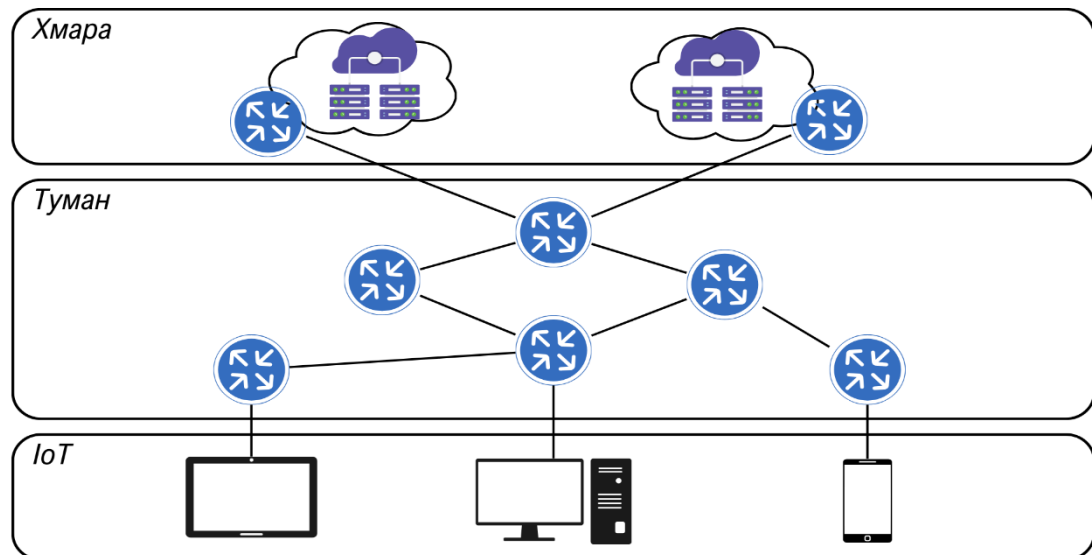


Рисунок 1.2 – Ландшафт туманних вичеслень

Високий географічний розподіл. Пристрої IoT дуже географічно розподілені. Ось чому потрібен перехід від централізованої обробки в хмарі до децентралізованої обробки в умовах туману.

Наявність великомасштабних сенсорних мереж. Широкомасштабні сенсорні мережі, що взаємодіють з осередками туману, є одним із ключових сценаріїв парадигми обчислення туману. Ці датчики здатні надсилати запити завдань до осередків туману. Залежно від доступних ресурсів туманних комірок, він або сам обробляє запити, або передає їх іншим туманним коміркам для подальшої обробки.

Підтримка мобільності. Окрім широкого розповсюдження пристроїв Інтернету речей, слід враховувати мобільність кінцевих пристроїв, які беруть участь. Рухомі пристрої вимагають динамічної реструктуризації топології мережі відповідно до постраждалих пристроїв у ландшафті туману. Це можливе через динамічний ієрархічний ландшафт туману, який дозволяє реструктурувати топологію. Для підтримки високої мобільності обчислювальний ландшафт туману повинен мати можливість спілкуватися з мобільними пристроями за допомогою технологій мобільного зв'язку, наприклад, LISP11.

Неоднорідність пристроїв. Різноманітні пристрої IoT, підключені до мережі, не мають стандартизованих функцій, інтерфейсів або типів розгортання. Отже, під час обробки запитів у туманному ландшафті необхідно враховувати багато різнорідних пристроїв IoT. Ландшафт туману спрямований на забезпечення зв'язку між різними типами пристроїв IoT за допомогою стандартизованого зв'язку та віртуалізації ресурсів.

Будучи нещодавно дослідженою темою, туманні обчислення стикаються з відсутністю методології та конкретних рішень. Отже, ще потрібно багато дослідницьких зусиль, щоб забезпечити характеристики, як зазначено вище. Важливою асоціацією, яка підтримує та поширює бачення туманних обчислень, є Open Fog Consortium¹². Принцип Open Fog Consortium полягає у збереженні відкритої обчислювальної архітектури для туману, щоб забезпечити спільні дослідження з кількома організаціями. Співпраця різнорідних організацій, що спеціалізуються в різних областях, є вирішальним моментом щодо майбутнього обчислень у тумані.

Fog computing часто згадується як технологія забезпечення для різних застосувань у різноманітних галузях, наприклад, охорона здоров'я, доповнена реальність, кешування та попередня обробка. Будучи в змозі забезпечити багатообіцяючі вдосконалення, які виникають завдяки цій парадигмі, ще потрібно вирішити багато проблем. Вирішальні проблеми в дослідженнях включають забезпечення ресурсами, розміщення послуг, безпеку та надійність, мінімізацію енергії, стандартизацію та моделі програмування.

Забезпечення ресурсами включає ефективне управління обчислювальними, зберігаючими та мережевими ресурсами в розподіленому середовищі fog computing. Розміщення послуг відноситься до оптимального розподілу задач між вузлами мережі для досягнення мінімальної затримки та високої продуктивності. Безпека та надійність є критично важливими для захисту даних та забезпечення стабільної роботи систем. Стандартизація сприяє забезпеченню сумісності між різними пристроями та платформами, що є необхідним для широкого впровадження технології. Моделі програмування полегшують розробку додатків для fog computing, забезпечуючи інструменти та абстракції для зручного створення розподілених систем.

Вирішення цих проблем дозволить повною мірою реалізувати потенціал fog computing, що сприятиме розвитку інноваційних застосувань у різних галузях.

Крім туманних обчислень, розвинулися дві дещо різні парадигми, які називаються Mobile Edge Computing (MEC) і Mobile Cloud Computing (MCC).

1.7 Мобільні периферійні обчислення

MEC базується на подібній ідеї, як туманні обчислення, але робить акцент на мобільних пристроях на межі мережі. Ключовими функціями всього середовища MEC є розподіл завдань між мобільними пристроями та перенесення завдань у хмару. Мобільні пристрої можуть спілкуватися між собою та отримувати запити завдань один від одного. Крім того, так звані

сервери MEC розташовані на краю мережі, відповідальні за підключення мобільних пристроїв до хмари. У цьому підході межа мережі є межею мережі радіодоступу (RAN). Отже, базові станції RAN представляють сервери MEC, тобто осередки туману в туманних обчисленнях. Ці сервери MEC належать телекомунікаційним компаніям і здатні обробляти запити завдань. Через те, що більша частина зв'язку відбувається в RAN, основними технологіями зв'язку в MEC є стільниковий зв'язок поколінь 3G, 4G і 5G.

Однією з основних рушійних сил, що підтримує дослідження в області MEC, є Європейський інститут стандартів телекомунікацій (ETSI). Ця рушійна сила, очевидно, пов'язана з інтенсивним використанням телекомунікаційних ресурсів і подальшими можливостями для телекомунікаційної галузі.

Підсумовуючи, головною метою MEC є зменшення затримки мережі та, таким чином, покращення якості мобільних додатків, наприклад, доповненої та віртуальної реальності, потокового відео на вимогу та мобільних ігор.

MCC виник на основі вже добре вивчених технологій мобільних обчислень і хмарних обчислень. Рушійною силою цієї технології є багате на ресурси середовище хмарних обчислень у поєднанні з поширенням розумних мобільних пристроїв. Хоча обчислювальна потужність сучасних мобільних пристроїв зростає, досі немає задовільного рішення щодо тривалості роботи батареї та ємності зберігання даних. Отже, ресурсомісткі завдання, наприклад, обробка зображень і обробка природної мови, повинні оброблятися в хмарі, щоб заощадити енергію та ємність зберігання. Однак, оскільки мобільні телефони з невикористаними процесорами є повсюдним, ефективне використання цих ресурсів має важливе значення. Наразі величезна кількість обчислювальної потужності витрачається, нехтуючи неактивними процесорами мобільних пристроїв. Зокрема, у певні періоди часу, наприклад, вночі, багато ресурсів можна використовувати без конфліктів.

Загальна ідея MCC полягає в тому, щоб розвантажити інтенсивні обчислювальні завдання, зменшити енергоспоживання та заощадити простір

для зберігання шляхом розповсюдження даних у хмару, мобільну хмару або потужні пристрої IoT.

Існує три різні типи досліджуваних середовищ МСС. По-перше, мобільний пристрій розвантажує запит завдання в хмару та отримує у відповідь рішення. По-друге, кілька мобільних пристроїв утворюють так звану мобільну хмару та обробляють запити завдань, що надходять від інших мобільних пристроїв, які цього потребують. По-третє, багатий на ресурси пристрій IoT, розташований на межі мережі, обробляє запити завдань мобільних пристроїв і повертає рішення. Далі останній тип розглядається, коли йдеться про МСС, оскільки це найбільш пов'язаний тип щодо парадигми туманних обчислень.

Сценарії застосування МСС включають мобільну охорону здоров'я, мобільну комерцію та мобільне навчання. Ці різноманітні сценарії застосування зосереджені на різних аспектах парадигми МСС. У сценарії мобільної охорони здоров'я дані, зібрані мобільними пристроями, відіграють вирішальну роль, оскільки дані можуть містити інформацію, що рятує життя, яку потрібно миттєво обробити. У мобільній комерції, з іншого боку, нові моделі зв'язку та можливості досягнення є найважливішими перспективами.

1.8 Програмні фреймворки

Фреймворк програмного забезпечення представляє базову структуру для розробки додатків і виконання послуг у певному програмному середовищі. Багаторазова базова структура служить для того, щоб дозволити користувачеві зосередитися на певному завданні, яке потрібно розробити, замість впровадження основ середовища. Фреймворки програмного забезпечення часто є розширюваними та конфігурованими відповідно до індивідуальних проблем, які користувачі хочуть вирішити за допомогою своїх програм.

Більш детально, фреймворки програмного забезпечення можна розділити на фреймворки білої та чорної скриньок. Фреймворки чорного

ящика – це готові до використання фреймворки, які не потрібно розширювати або вдосконалювати для виконання. Фреймворки білої скриньки, з іншого боку, вимагають налаштувати шляхом розширення певних інтерфейсів, щоб виконувати фреймворк у вибраному середовищі. Найпоширеніші фреймворки є сумішню фреймворків білої та чорної скриньок. У цій дисертації ця суміш розглядається, коли йдеться про фреймворки. У наступних параграфах представлено відповідні програмні структури в середовищах IoT.

1.9 Надання ресурсів

Забезпечення ресурсами є важливою темою в багатьох різноманітних областях досліджень. Очевидно, що важливо використовувати наявні ресурси якомога ефективніше. У цьому контексті ефективність пов'язана з різними цілями, наприклад, оптимізацією витрат, енергії, часу роботи та використання ресурсів. Надання ресурсів – це процедура організації, розподілу, звільнення та моніторингу доступних системних ресурсів. Ці згадані дії мають вирішальне значення для забезпечення ефективного управління ресурсами з урахуванням якості обслуговування, тобто надання ресурсів. Отже, необхідно враховувати не тільки залежний від часу попит, але також необхідно відстежувати показники QoS і перехресно перевіряти за допомогою SLA. Залежно від коливань навантаження на систему процедура надання ресурсів змінюється за складністю. Зі збільшенням коливань процедура стає складнішою, оскільки ресурси потребують все більшої адаптації. У наступних параграфах описані найважливіші аспекти забезпечення ресурсами.

Моніторинг. Системний моніторинг необхідний для відстеження статусу запущених служб і програмного середовища в цілому, наприклад, використання ЦП і оперативної пам'яті, зміни топології, збої. Зібрані дані зберігаються в базі даних для подальшої обробки. Для моніторингу повного програмного середовища модель системи є важливою. Вкрай важливо знайти

необхідний рівень абстракції для моделювання гетерогенних компонентів системи.

Вимірювання та оркестрація. Обґрунтування та оркестровка – це процес, у якому аналізується інформація моніторингу системи та обчислюється план надання ресурсів. План надання ресурсів визначає, де розгортаються конкретні служби та в яких службах обробляються вхідні запити завдань.

Розподіл і звільнення ресурсів. Відповідно до розрахованого плану забезпечення ресурсами, описаного в попередньому параграфі, ресурси туману розподіляються та звільнюються.

Процедура надання ресурсів може здійснюватися проактивним або реактивним способом. Проактивний сценарій передбачає періодичну перевірку стану системи, і відповідно до даних моніторингу здійснюється надання ресурсів. Часто механізми прогнозування застосовуються для прогнозування майбутніх потреб у ресурсах і для того, щоб діяти до того, як система буде навантажена. З іншого боку, реактивний сценарій надання безпосередньо стосується таких подій, як збої, поява нових пристроїв, сторожових подій тощо, і відповідно обробляє надання. Наприклад, сторожовий компонент запускає подію перевантаження, що означає, що використання ЦП конкретної служби перевищує визначений поріг, і система реагує, розгортаючи більше екземплярів цієї самої служби.

Крім того, процедура надання ресурсів може здійснюватися за кількома різними підходами. Залежно від задачі оптимізації, що виникає в результаті моделі системи, мети оптимізації та подальших обмежень вибирається адекватний підхід. Задача оптимізації може бути сформульована в термінах динамічного програмування, точніше лінійного програмування, і може бути вирішена або точними математичними методами, або евристичними алгоритмами. Евристики – це алгоритми, близькі до реальних проблем, які шукають майже оптимальні рішення. Щоб покращити рішення евристики, часто сценарій проблеми та середовище потребують уточнення й опису більш

детально. Прикладами евристичних алгоритмів є жадібні алгоритми та локальний пошук.

У середовищі туманних обчислень надання ресурсів є ще складнішим порівняно з хмарним середовищем, оскільки розподілені різномірні пристрої повинні бути забезпечені з мінімальною затримкою та часом виконання завдань. Крім того, ландшафт обчислень туману містить динамічну ієрархію. Таким чином, ієрархія пристроїв може змінюватися під час виконання, тому вона недоступна на початку служби. Очевидно, що вже дуже добре вивчені підходи до надання ресурсів хмарних обчислень не можуть бути безпосередньо застосовані в ландшафті туманних обчислень. Залишається виділити загальну ідею цих підходів хмарних обчислень і використати їх для подальшої специфікації в умовах туману.

2 ДОСЛІДЖЕННЯ МЕТОДУ ПІДВИЩЕННЯ БЕЗПЕКИ В ПОЛІНГОВИХ МЕРЕЖАХ

2.1 Дослідження полігрових мереж

Полінгові системи, також відомі як системи впорядкованого опитування, – це масивна система обслуговування, що містить кілька черг і один сервісний пристрій. Пристрій адресує та обслуговує чергу за певним алгоритмом. Слово «полінгові» походить від англійського «polling» і описує метод керування лінією передачі даних, за якого центральний комп'ютер циклічно опитує периферійні термінали, щоб визначити, які дані передавати. Після завершення передачі даних канал стає доступним для сервісної програми, і хост переходить до наступного терміналу.

Вимірнювальні системи використовуються не лише в комп'ютерних і комунікаційних системах, а й в інших технічних областях, таких як виробництво та транспорт. Багато дослідників вивчали різні варіанти систем голосування, зосереджуючись на застосуванні нових технологій у кожен період. Наприклад, британська бавовняна промисловість використовувала однорядну модель стовпів для вирішення проблем, які виникали, коли обслуговуючий персонал відвідував фабрики. Також люди використовували модель двох черг для аналізу управління трафіком. З розвитком комп'ютерних мереж зв'язку були проведені широкомасштабні дослідження схем опитування для передачі даних від периферійних терміналів до центральних комп'ютерів. Протокол передачі маркерів локальної мережі (LAN) відновив схему опитування, дозволивши використовувати спеціальні біти для передачі даних у мережах Token Ring і Token Bus.

Полінгові системи є ключовим елементом забезпечення ефективної роботи багатьох сучасних технологічних систем. Їх використання оптимізує розподіл ресурсів і скорочує час простою обладнання. Наприклад, у

виробничих системах механізми політінгу можуть допомогти координувати роботу різних виробничих ліній і забезпечити своєчасне обслуговування та ремонт. У транспортних політінгові системи використовується для контролю руху та забезпечення безпеки дорожнього руху. У майбутньому розвиток таких технологій, як Інтернет речей (IoT) і штучний інтелект (AI), може ще більше вдосконалити системи голосування, зробивши їх більш адаптивними та ефективними.

Наприкінці 1990-х років системи опитування з додатковими функціями контролю, такими як пріоритет і часові обмеження, використовувалися в міських мережах (MAN), високошвидкісних локальних мережах, цифрових мережах інтегрованих послуг (ISDN) і протоколах доступу до наземних каналів мобільного зв'язку. Інтернет і мережі супутникового мовлення.

За останні роки бездротові мережі передачі даних стали одним із ключових напрямків розвитку мережевої індустрії. Сучасні мережі надають користувачам широкий спектр послуг, таких як електронна пошта, голосові та відеоповідомлення, доступ у реальному часі до віддалених баз даних, служби новин та інші послуги. На базі цієї мережі створені дистанційне навчання, телемедицина, телеконференції, пошукові системи, соціальні мережі, онлайн-ігри, інтернет-магазини, каталоги товарів і послуг, електронні носії тощо.

Різке збільшення кількості комп'ютерних мереж і успішний розвиток дротових і бездротових методів зв'язку супроводжується постійними змінами в мережевих технологіях, спрямованих на підвищення швидкості і надійності мереж, а також можливості комплексної передачі різних типів інформації.

Обмін даними, голосом і відео. Використання бездротової технології дозволяє швидко і з мінімальними витратами об'єднати віддалені локальні мережі і робочі станції в мережу передачі даних, забезпечивши користувачам локальної мережі віддалений доступ до мережі Інтернет. Політінгові системи також використовуються для оцінки характеристик бездротових мереж. Тепер можна з упевненістю сказати, що вони стали потужним інструментом для аналізу ефективності та продуктивності в багатьох важливих сферах.

Вони також використовуються в ситуаціях, коли кілька користувачів змагаються за доступ до одного ресурсу, який одночасно може використовувати лише один із них. Сучасні дослідження та вдосконалення в цій галузі продовжують розширювати можливості систем голосування, роблячи їх більш адаптивними та ефективними у вирішенні нових завдань.

Найпростішим видом полінових систем є так звана класична система поліну, яка складається з одного обслуговуючого пристрою та кількох ($N > 1$) черг з необмеженою кількістю місць для очікування. У кожен чергу Q_i ($i = 1, 2, \dots, N$) надходять заявки з інтенсивністю λ_i . Сервісний пристрій циклічно проходить чергу, обслуговуючи резерв запитів. За класифікацією Кендалла це система масового обслуговування M/G/1. Ця класична система інтенсивно вивчалася багатьма дослідниками, і існує багато літератури, присвяченої її аналізу.

Сучасні технологічні досягнення в галузі інформаційних і комунікаційних технологій призвели до створення більш досконалих моделей поліну з новими і додатковими функціями, недоступними в класичних системах. Дослідження та розробка нових систем із розширеною функціональністю, здатних задовольняти різноманітні вимоги до інформаційних і комунікаційних систем та інших технологічних галузей, стають головними науковими проблемами.

Завдяки постійному вдосконаленню сучасні полінові системи містять більш складні алгоритми керування, які можуть оптимізувати процеси обслуговування та підвищити ефективність і надійність системи. Наприклад, методи пріоритетного обслуговування або застосування обмежень часу очікування можуть значно покращити продуктивність системи в умовах високого навантаження. Ці вдосконалення роблять системи опитування важливим інструментом для аналізу та оптимізації в різних галузях, таких як телекомунікації, виробництво та транспорт.

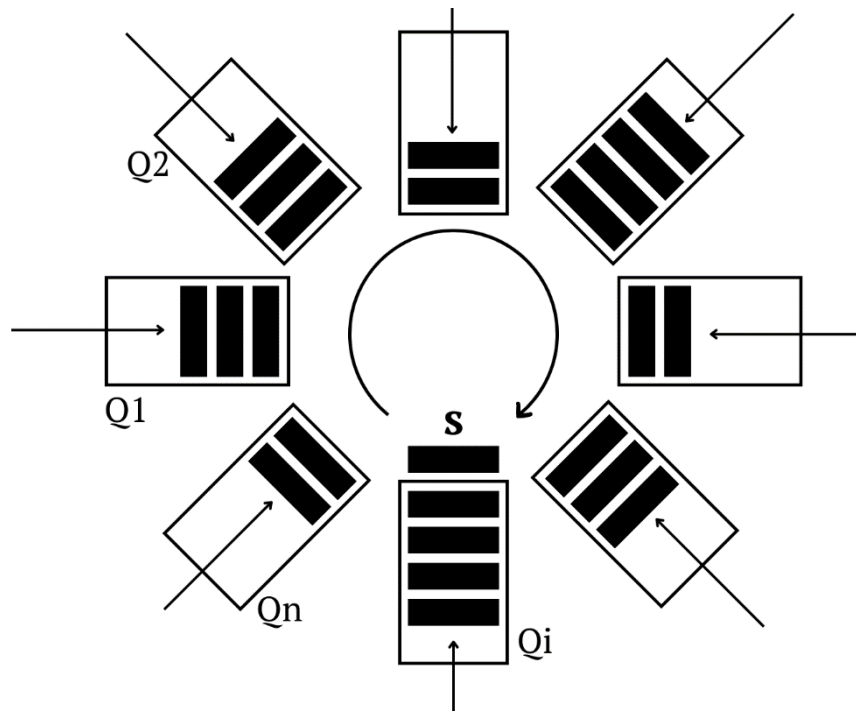


Рисунок 2.1 – Система полігової мережі

Основними характеристиками полігових систем є трафік поповнення черги, місткість черги (кількість місць очікування), дисципліна обслуговування заявки, маршрутизація обслуговуючого пристрою, режим обслуговування черги, час перемикання обслуговуючого пристрою між чергами, кількість обслуговуючих пристроїв і кількість черг. Шлях поповнення, методи обслуговування та час перемикання мають найбільший вплив на продуктивність системи, але вони не можуть повністю визначити продуктивність системи. Продуктивність також залежить від інших згаданих параметрів.

У більшості досліджень систем опитування передбачається, що потік додатків, який заповнює чергу, розподілений за Пуассоном. Це вірно в багатьох ситуаціях, таких як телефонні дзвінки та дорожньо-транспортні пригоди. Системи опитування з пуассонівськими вхідними потоками легко аналізувати.

Однак у реальному житті деякі системи не підходять для цього припущення, особливо у випадку великого трафіку (наприклад, передачі

голосу чи відео). У системах опитування часто припускають, що запити надходять із зовнішнього необмеженого джерела, але в деяких випадках це припущення нереалістичне, наприклад, у сервісних центрах і відділах технічної підтримки. Модель опитування з безперервним потоком додатків детально обговорюється в дослідженні, а результати порівнюються з результатами дискретної моделі.

Сучасні дослідження систем опитування громадської думки спрямовані на врахування всіх аспектів їх роботи та підвищення ефективності. Наприклад, розробка алгоритмів, які дозволяють враховувати пріоритет запиту або обмежувати час очікування, може значно підвищити продуктивність системи. Це робить системи опитування важливим інструментом для оптимізації процесів у таких галузях, як телекомунікації, виробництво та транспорт. Удосконалення та адаптація моделей опитування до реальних ситуацій забезпечує їх ефективне застосування у все більш широкому діапазоні завдань.

Ємність черги означає кількість місць, які очікують на обслуговування. У більшості систем опитування передбачається, що черга має необмежену ємність, але в деяких випадках, наприклад у транспортних і виробничих системах, ємність має бути обмежена. Існують також системи, черги яких мають одиничну місткість, тобто кожна черга може містити лише одну заявку за раз.

Маршрутизація службового пристрою або порядок опитування визначає, як сервісний пристрій отримує доступ до черги обслуговування. Маршрути можуть бути статичними і динамічними. При статичній маршрутизації вибір черги обслуговування не залежить від поточного стану системи, а правила вибору черги залишаються незмінними під час роботи системи. Динамічна маршрутизація залежить від стану системи та вибирає черги для обслуговування на основі доступної інформації, такої як довжина черги або наявність повідомлень високого пріоритету. Кругова маршрутизація – це тип маршрутизації сервісного пристрою. Відповідно до цього типу після завершення обслуговування однієї черги обслуговуючий пристрій переходить

до наступної черги в порядку проходження. Це означає, що після обслуговування черги Q_i обладнання переходить до черги Q_{i+1} , потім до черги Q_{i+2} і так далі до черги Q_N , після чого повертається до черги Q_1 і цикл повторюється.

Періодична маршрутизація задається таблицею полігону

$$\{T(1), T(2), T(3), \dots, T(M)\}. \quad (2.1)$$

Довжини $M (M \geq N), T(i) \in \{1, 2, 3, \dots, N\}, I = 1, M$, обслуговуюче обладнання відвідує черги один по одному. При цьому передбачається, що таблиця полігону містить номері всіх черг в системі.

Пріоритетна маршрутизація полягає в тому, що система має різні черги з різними пріоритетами. При цьому обслуговування може відбуватися лише в тому випадку, якщо у більш пріоритетних черг відсутні заявки.

Режим обслуговування черг, або спосіб їхнього обслуговування, визначає, скільки заявок буде обслужено обслуговуючим обладнанням під час кожного відвідування черги. Існують чотири основні типи режимів:

- в режимі "вичерпний" обслуговуюче обладнання працює, поки черга не порожніє;
- у режимі "шлюзовий" обслуговуються тільки ті заявки, які були у черзі на початку процесу обслуговування, нові заявки обробляються лише в наступному циклі;
- в обмеженому режимі обслуговуються обмежена кількість заявок за відвідування черги;
- обмежений за часом режим означає, що обслуговуюче обладнання працює, поки не мине визначений проміжок часу або поки черга не стане порожньою, залежно від того, що настане перше.

Розробники систем іноді використовують комбіновані підходи до обслуговування черг для покращення продуктивності. Це означає, що в системі може бути декілька методів обслуговування, і кожна черга може мати

свою власну стратегію, залежно від її пріоритету. Наприклад, для полігрових систем з обмеженим режимом обслуговування можна використовувати нейронні мережі для визначення оптимальної кількості заявок.

Дисципліна обслуговування черг визначає, у якому порядку обслуговуватимуться заявки, що знаходяться у черзі. Найпоширенішою дисципліною є "перший прийшов, перший вийшов" (МІМО). Оптимізація цієї стратегії може значно підвищити продуктивність системи, не потребуючи додаткових ресурсів.

Час перемикання від однієї черги до іншої, у деяких випадках, є миттєвим. Збільшення кількості обслуговуючого обладнання є одним із шляхів оптимізації систем полігру, особливо в мобільних мережах, де це дозволяє підвищити якість передачі даних. Узагальнено, параметри систем полігру включають в себе не лише кількість черг та обслуговуючих обладнань, але й методи обслуговування та параметри потоків даних.

2.2 Дослідження оверлейних мереж

Оверлейні мережі, зокрема VPN, є потужним інструментом в сучасному цифровому світі. Вони створюють безпечний тунель між двома або більше вузлами мережі через непрозорий канал Інтернету. Основна їхня перевага полягає в забезпеченні конфіденційності, аутентифікації та цілісності даних, що передаються через мережу. Це особливо важливо в умовах зростання кількості кіберзлочинів і порушень безпеки в Інтернеті.

Оверлейні мережі являють собою віртуальні або логічні мережі, які функціонують поверх фізичної мережевої інфраструктури. Вони забезпечують додатковий рівень абстракції, дозволяючи створювати гнучкі, масштабовані та безпечні мережеві рішення, що є критично важливими для сучасних технологій, таких як Інтернет речей (IoT).

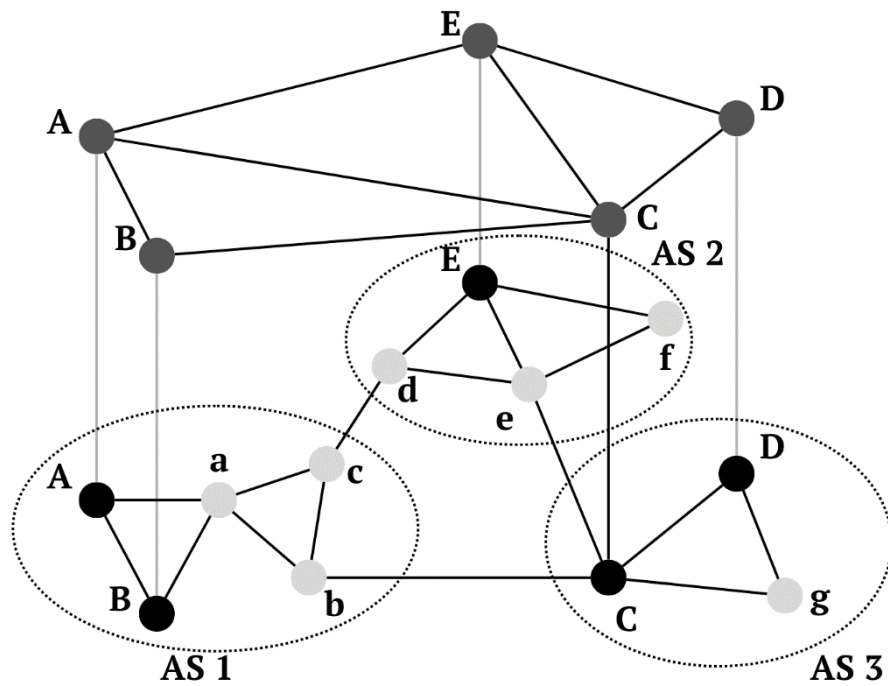


Рисунок 2.2 – Система оверлейної мережі

Основні концепції оверлейних мереж включають наступні аспекти:

Віртуальні канали зв'язку. Однією з ключових концепцій оверлейних мереж є використання віртуальних каналів зв'язку. Ці канали створюються поверх існуючих фізичних з'єднань, що дозволяє логічно об'єднувати різні вузли мережі, незалежно від їхнього фізичного розташування. Віртуальні канали забезпечують більш ефективне використання мережевих ресурсів, оскільки вони дозволяють динамічно налаштовувати маршрути трафіку відповідно до поточних умов мережі.

Абстракція від фізичної інфраструктури. Оверлейні мережі забезпечують рівень абстракції, що дозволяє відокремити логічну топологію мережі від її фізичної інфраструктури. Це означає, що мережеві адміністратори можуть створювати та керувати віртуальними мережами незалежно від фізичних обмежень, таких як розташування мережевих пристроїв або типи з'єднань. Абстракція дозволяє також спростити управління мережею, оскільки всі налаштування можуть бути здійснені на логічному рівні.

Оптимізація маршрутизації. Однією з важливих функцій оверлейних мереж є оптимізація маршрутизації трафіку. Завдяки використанню віртуальних каналів та абстракції від фізичної інфраструктури, оверлейні мережі можуть використовувати більш ефективні алгоритми маршрутизації, що дозволяють зменшити затримки, підвищити пропускну здатність та забезпечити більш надійний зв'язок між вузлами. Це особливо важливо для IoT, де кількість підключених пристроїв може бути дуже великою, а вимоги до якості зв'язку – високими.

Підвищення безпеки. Оверлейні мережі також надають можливості для підвищення безпеки мережевого трафіку. Завдяки використанню шифрування та інших методів захисту даних, оверлейні мережі можуть забезпечувати конфіденційність та цілісність переданих даних. Крім того, логічна ізоляція віртуальних мереж дозволяє запобігати несанкціонованому доступу до критичних ресурсів та зменшувати ризики атак на мережеву інфраструктуру.

Масштабованість. Оверлейні мережі забезпечують високу масштабованість, що є критично важливим для IoT. Завдяки можливості динамічно додавати нові віртуальні вузли та канали, оверлейні мережі можуть легко адаптуватися до зростаючих вимог мережевої інфраструктури. Це дозволяє забезпечувати стабільну роботу мережі навіть при значному збільшенні кількості підключених пристроїв.

Гнучкість та адаптивність. Оверлейні мережі відрізняються високою гнучкістю та адаптивністю. Вони можуть бути легко налаштовані для задоволення специфічних вимог різних застосувань, що робить їх ідеальними для використання в IoT, де можуть бути різноманітні вимоги до мережевої інфраструктури. Наприклад, оверлейні мережі можуть бути налаштовані для підтримки різних типів трафіку, таких як низьколатентний трафік для реального часу або високонадійний трафік для критичних додатків.

Підтримка різних протоколів. Оверлейні мережі підтримують різні мережеві протоколи, що дозволяє їм інтегруватися з існуючими мережевими рішеннями та забезпечувати взаємодію між різними типами пристроїв. Це є

важливою перевагою для IoT, де можуть використовуватися різні стандарти та протоколи для зв'язку між пристроями.

Оверлейні мережі, як ключова технологія для підтримки Інтернету речей (IoT), потребують спеціалізованих протоколів та алгоритмів для забезпечення ефективного функціонування. Зважаючи на унікальні вимоги IoT, такі як висока кількість підключених пристроїв, різноманітність даних, що передаються, і необхідність в реальному часі реагувати на події, протоколи та алгоритми для оверлейних мереж повинні відповідати цим специфічним викликам. Одним із ключових аспектів протоколів для оверлейних мереж є ефективне управління маршрутизацією. У контексті IoT це означає забезпечення надійної та швидкої доставки даних від численних сенсорів і пристроїв до центральних вузлів обробки. Алгоритми маршрутизації, які використовуються в оверлейних мережах IoT, повинні враховувати обмежені ресурси пристроїв, такі як енергоефективність і обмежена пропускна здатність. Наприклад, протокол RPL (Routing Protocol for Low-Power and Lossy Networks) є популярним вибором для IoT через його здатність забезпечувати ефективну маршрутизацію в мережах з низькою потужністю та високою втратою пакетів. RPL будує спрямовані ациклічні графи (DAG) для маршрутизації даних, що дозволяє динамічно адаптуватися до змін у топології мережі.

Крім маршрутизації, важливу роль відіграють протоколи керування трафіком. У випадку оверлейних мереж для IoT ці протоколи повинні забезпечувати ефективний розподіл трафіку між різними віртуальними каналами, що дозволяє зменшити затори та підвищити загальну продуктивність мережі. Для цього використовуються алгоритми балансування навантаження, які розподіляють трафік на основі поточних умов мережі, таких як завантаженість вузлів і пропускна здатність каналів. Алгоритми, засновані на техніках машинного навчання, можуть прогнозувати майбутні навантаження та відповідно коригувати маршрутизацію та розподіл ресурсів. Ще одним важливим аспектом є забезпечення безпеки в оверлейних мережах IoT. Протоколи шифрування та автентифікації є критично важливими для

захисту даних, що передаються через оверлейні мережі. Протоколи, такі як DTLS (Datagram Transport Layer Security), забезпечують захищений транспорт даних для додатків, що використовують ненадійні транспортні протоколи, такі як UDP. DTLS дозволяє пристроям IoT обмінюватися даними з використанням шифрування, що гарантує конфіденційність та цілісність переданих повідомлень. Крім того, протоколи автентифікації забезпечують ідентифікацію пристроїв та запобігають несанкціонованому доступу до мережі. У контексті оверлейних мереж важливо також враховувати питання масштабованості. Протоколи та алгоритми повинні бути здатні підтримувати зростаючу кількість підключених пристроїв без значних втрат продуктивності. Одним із підходів до вирішення цього питання є використання децентралізованих алгоритмів, які дозволяють кожному вузлу мережі самостійно приймати рішення на основі локальної інформації. Це зменшує навантаження на центральні вузли та підвищує загальну стійкість мережі. Протоколи управління даними також відіграють важливу роль в оверлейних мережах IoT. Вони забезпечують ефективний збір, обробку та зберігання даних від численних сенсорів та пристроїв. Одним із популярних протоколів для цієї мети є MQTT (Message Queuing Telemetry Transport), який розроблений для надійної передачі даних у мережах з обмеженими ресурсами. MQTT використовує модель "публікація-підписка", що дозволяє пристроям ефективно обмінюватися даними без постійного підключення до центрального сервера. Окрім протоколів, важливу роль відіграють алгоритми управління ресурсами. У оверлейних мережах IoT вони відповідають за оптимальний розподіл обчислювальних та комунікаційних ресурсів між різними пристроями та додатками. Алгоритми управління ресурсами, засновані на концепціях розподілених обчислень та віртуалізації, дозволяють ефективно використовувати наявні ресурси та забезпечувати високу продуктивність мережі. Таким чином, протоколи та алгоритми для оверлейних мереж в IoT є ключовими елементами, що забезпечують надійне, безпечне та ефективне функціонування мережі. Вони враховують специфічні вимоги IoT, такі як

обмежені ресурси, висока масштабованість, безпека та ефективне управління трафіком, що дозволяє забезпечувати стабільну роботу мережі навіть у умовах значного зростання кількості підключених пристроїв.

2.3 Розробка моделі поліінгової системи з повним режимом обслуговування

Метою методу аналізу середніх показників часу системи в будь-який момент часу. Тому, окрім середньої тривалості перебування в черзі обслуговуючим обладнанням, потрібно знати також середній залишковий час перебування в черзі.

Метод аналізу середніх передбачає, що якщо режим обслуговування є вичерпним, то тривалість відвідування V_i дорівнює сумі часу перемикання обслуговуючого обладнання до черги S_i та часу обслуговування всіх заявок у черзі. Якщо ж у системі використовується шлюзовий режим обслуговування, то V_i дорівнює сумі часу підключення обслуговуючого обладнання до наступної $(i + 1)$ -ї черги S_{i+1} і часу обслуговування заявок у черзі. З такої інтерпретації тривалості відвідування випливає, що кількість заявок у черзі дорівнює нулю на момент початку опитування черги (у випадку шлюзового режиму обслуговування) або на момент, коли обладнання, що обслуговує, залишає чергу (у випадку повного режиму обслуговування).

Розглянемо поліінгову систему з вичерпним режимом обслуговування. Спостереження ведеться за довільною заявкою з моменту її вступу в чергу, припустимо, в i -ту. Перш ніж отримати обслуговування, ця заявка чекатиме обслуговування всіх заявок, які вже перебувають у черзі на момент її вступу, а також залишковий час циклу. Обслуговуюче обладнання, в момент вступу заявки, може або обслуговувати заявку в i -тій черзі з ймовірністю ρ_i , тоді заявка повинна чекати завершення поточного обслуговування, або перемикатися до i -ї черги з ймовірністю S_i/C , тоді час очікування збільшується на залишковий час перемикання, або обслуговувати іншу чергу,

і тоді обслуговування заявки затримується до повернення обладнання до i -ї черги. Тривалість цього періоду дорівнює сумі залишкового часу до завершення $(i + 1)$ -го до $(N - 1)$ -го періодів і часу перемикання до i -ї черги. Оскільки при вичерпній дисципліні обслуговування черга після обслуговування стає порожньою, кількість заявок в i -тій черзі у будь-який момент часу $(i + 1)$ -го до (j) -го періодів дорівнює кількості заявок, що надійшли протягом цього періоду, оскільки розподіл тривалості цього періоду збігається з розподілом залишкового часу.

3 РОЗРОБКА МЕТОДУ ОРГАНІЗАЦІЇ ТУМАННИХ ОБЧИСЛЕНЬ В ПОЛІНГОВИХ МЕРЕЖАХ

3.1 Розробка архітектури моделі

Туманні обчислення, як розширення концепції хмарних обчислень, пропонують обробку даних ближче до місця їх виникнення, тобто на периферії мережі. Це дозволяє зменшити затримки, знизити навантаження на магістральні канали та забезпечити більш швидкий доступ до необхідних обчислювальних ресурсів. Одним із перспективних підходів до реалізації туманних обчислень є їх інтеграція в полінгові мережі (polling networks).

Полінгові мережі представляють собою структури, в яких вузли мережі періодично опитуються для збору та обробки інформації. Такий підхід дозволяє ефективно розподіляти ресурси між вузлами та забезпечувати високий рівень надійності та відмовостійкості системи.

Цей розділ дипломної роботи присвячено розробці методу організації туманних обчислень в полінгових мережах. У ньому розглядаються основні принципи та підходи до інтеграції туманних обчислень в полінгові мережі, аналізуються їх переваги та недоліки, а також пропонуються методи підвищення ефективності таких систем. На рисунку нижче наведено загальну структуру організації туманних обчислень в полінгових мережах.

На рисунку 3.1 зображена архітектура запропонованої моделі. Модель складається з наступних елементів:

- агенти, які збирають дані;
- вузлові агенти, які відповідальні за транспортування даних.

В попередньому розділі було описано модель полінгової системи, яка складається з агентів та вузлових агентів. Ця архітектура запропонованої моделі демонструє важливість розподіленої обробки даних і координації між різними компонентами системи.

Але в реальності вузлові агенти можуть бути реконфігуровані в звичайні агенти та навпаки. Розподілення та класифікація агентів необхідні для мінімізації витрати енергії. Бо доступ до мережі вимагає багато ресурсів. Ідея полягає в тому, що кількість вузлових агентів визначається частотою виникнення подій та кількістю даних, які необхідні для транспортування. В такому випадку число агентів можна залежитиме від інтенсивності надходження подій, активної черги та їх пріоритетності та обсягу.

Вирахувати необхідну кількість агентів можна за допомогою:

$$N_v = N_a * \lambda, \quad (3.1)$$

де: N_v – кількість вузлових агентів, необхідних системі;

N_a – загальна кількість агентів, доступних в системі;

λ – інтенсивність, з якою надходять нові заявки в систему.

В свою чергу для обрахування інтенсивності слід застосувати формулу:

$$\lambda = \frac{1}{\tau_s}, \quad (3.2)$$

де: τ_s – тривалість обробки заявки.

Розрахунок необхідної кількості вузлових агентів є важливим кроком у плануванні ресурсів системи. Знання точного числа агентів дозволяє ефективно розподіляти завдання і забезпечувати своєчасну обробку заявок, що надходять. Це особливо критично в умовах, де час реакції є важливим фактором, наприклад, у службах підтримки клієнтів або в логістичних компаніях.

Інтенсивність надходження нових заявок (λ) визначає, як часто система отримує нові завдання для обробки. Відповідно, правильний розрахунок цієї інтенсивності допомагає уникнути перевантаження системи та забезпечити її стабільну роботу. Використання формули (3.2) дозволяє з легкістю обчислити

інтенсивність на основі середньої тривалості обробки заявки (τ_s), що є важливим для ефективного управління ресурсами.

В свою чергу необхідно ввести поняття максимально допустимого ступеня завантаженості вузлового агента при якому системі необхідно реконфігурувати себе та додавати ще одного вузлового агента. Назвемо цей параметр μ . Тоді для обрахування гранично допустимого значення скористаємось наступною формулою:

$$\varphi = (l_{aMax} + 1) * \tau_l, \quad (3.3)$$

де: φ – показник максимальної часової затримки;

τ_l – час обробки одного елемента;

l_{aMax} – кількість елементів на вузлі з найбільшою завантаженістю.

При досяганні умови $\varphi \geq \mu$ система розуміє необхідність створення ще одного додаткового вузла. Такий підхід забезпечує динамічну адаптацію системи до змінних навантажень, додаючи необхідних агентів, тим самим підтримуючи ефективність і запобігаючи вузьким місцям у продуктивності. Адаптивність і масштабованість є критично важливими для систем, які відчувають коливання попиту і потребують можливостей обробки в реальному часі. Завдяки попередньому обчисленню та моніторингу цих параметрів адміністратори можуть забезпечити, щоб система залишалася надійною і швидко реагувала на різні умови експлуатації.

Це особливо важливо в умовах високої конкуренції, де швидкість і ефективність обробки заявок можуть стати ключовими факторами успіху. Динамічне додавання агентів дозволяє уникнути затримок та забезпечити своєчасну реакцію на запити користувачів, що підвищує загальну продуктивність системи. Крім того, такий підхід дозволяє оптимально використовувати ресурси, що є важливим для зниження витрат і підвищення економічної ефективності.

Системи з високою адаптивністю можуть легко масштабуватись відповідно до потреб, що забезпечує гнучкість і стійкість у разі непередбачуваних навантажень. Це стає можливим завдяки регулярному моніторингу та аналізу показників продуктивності, що дозволяє оперативно реагувати на зміни і вчасно вносити корективи в конфігурацію системи.

Крім того, автоматизація процесу додавання агентів зменшує потребу в ручному втручанні, що знижує ризик людських помилок і підвищує загальну надійність системи. Завдяки цьому адміністратори можуть зосередитися на стратегічних завданнях, не витрачаючи час на постійний контроль і налаштування системи. Також такий підхід забезпечує повну автономність і, як результат, меншу кількість людино-годин, які варто витратити на обслуговування, контроль та налагодження системи.

Крім того, автоматизація процесу додавання агентів зменшує потребу в ручному втручанні, що знижує ризик людських помилок і підвищує загальну надійність системи. Завдяки цьому адміністратори можуть зосередитися на стратегічних завданнях, не витрачаючи час на постійний контроль і налаштування системи. Також такий підхід забезпечує повну автономність і, як результат, меншу кількість людино-годин, які варто витратити на обслуговування, контроль та налагодження системи.

В результаті, впровадження такого підходу сприяє підвищенню задоволеності користувачів, адже вони отримують швидкий і безперебійний доступ до сервісів. Це також позитивно впливає на репутацію компанії, оскільки стабільність і надійність її систем стають очевидними для всіх користувачів.

Для обчислення l_{aMax} необхідно для кожного вузла вирахувати l_{ai} :

$$l_{ai} = \tau_{li} * (l_i + 1), \quad (3.4)$$

де: l_{ai} – час очікування і-того вузла;

τ_{li} – час обробки одиниці даних на і-тому вузлі;

l_i – кількість одиниць даних, очікуючих в черзі на обробку.

Для того щоб ефективно визначити максимальне значення часу очікування, необхідно створити матрицю, яка міститиме значення для всіх вузлів системи. Кожен елемент цієї матриці відповідає часу очікування на певному вузлі, розрахованому за формулою (3.4).

$$l_{aMax} = \begin{bmatrix} l_{a1} \\ l_{a2} \\ l_{a3} \\ \vdots \\ l_{an} \end{bmatrix} \quad (3.5)$$

Після побудови матриці потрібно проаналізувати всі її елементи та знайти максимальне значення, що буде відповідати найбільшому часу очікування серед усіх вузлів. Це максимальне значення (l_{aMax}) є критичним показником для системи, оскільки воно визначає найгірший сценарій затримки, з яким може стикнутися система. Знання цього значення дозволяє приймати обґрунтовані рішення щодо оптимізації роботи вузлів і розподілу навантаження, щоб мінімізувати час очікування і покращити загальну ефективність системи.

Таким чином, створення матриці часу очікування і аналіз її максимального значення є ключовим етапом у процесі оптимізації і забезпечення стабільної роботи системи. Це допомагає виявити вузькі місця та розробити стратегії для їх усунення, що сприяє більш ефективному управлінню ресурсами і підвищенню продуктивності всієї системи.

Як результат, архітектура системи, як зображено на рисунку 3.2, відображає розширену мережу вузлових агентів, що працюють у взаємодії для оптимізації обробки даних і відповіді на запити. Ця архітектура віддзеркалює важливість розподілу завдань і координації між агентами, що дозволяє системі ефективно впоратися з великими обсягами роботи та забезпечити стабільну роботу навіть у високонавантажених умовах.

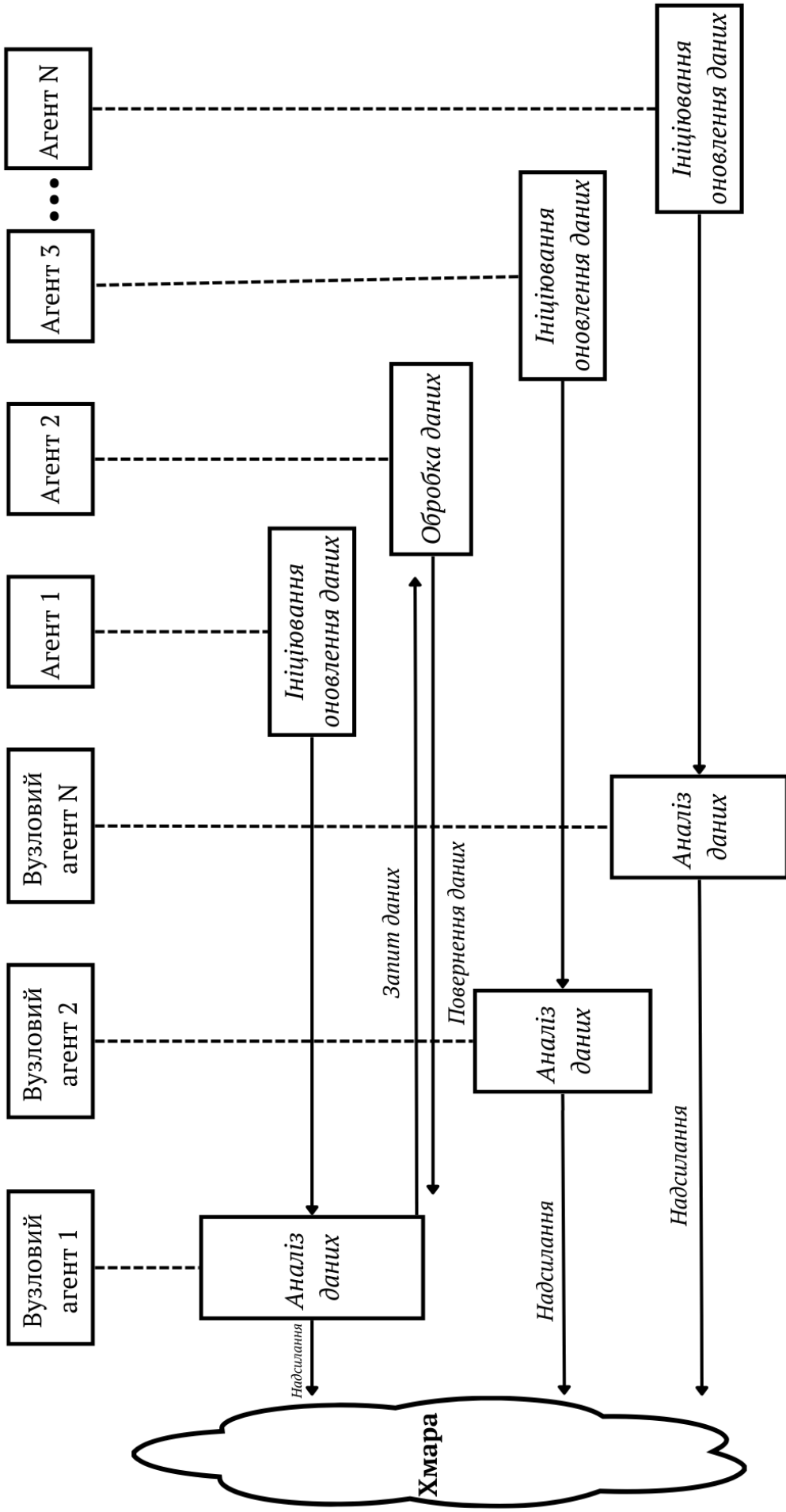


Рисунок 3.2 – Архітектура запропонованої моделі з декількома вузлами

В даному випадку є комбінація централізованої та денентралізованої систем. В рамках дослідження системи туманних обчислень в полігрових мережах важливим аспектом є ступінь її децентралізації. Система організована таким чином, що кожен вузловий агент має можливість самостійно відправляти дані до хмарових обчислювальних ресурсів, що дозволяє досягти гнучкості та розподіленості в управлінні і обробці інформації. В той же час звичайні агенти при базовій конфігурації не мають доступу ні до даних інших агентів, ні доступу до хмари. Цей підхід сприяє зменшенню залежності від централізованого вузла керування, що забезпечує високу стійкість до відмов і покращує швидкість відгуку системи на змінні умови.

Кожен вузловий агент у системі туманних обчислень має свої власні завдання та відповідальність за збір та передачу даних у хмару. Це означає, що навіть у випадку наявності кількох вузлових агентів кожен з них може функціонувати незалежно від інших, забезпечуючи неперервний потік інформації до централізованої обробки. Під таким підходом реалізації система стає більш масштабованою та гнучкою, спроможною адаптуватися до зростаючих потреб у обробці даних та вимог до швидкодії. Кожен агент відповідає за власну частину обчислювального процесу, що сприяє збільшенню загальної ефективності системи та зниженню загальних витрат на її підтримку.

3.2 Застосування оверлейних мереж

Розглянемо оверлейні мережі у контексті систем туманних обчислень, де агенти та вузлові агенти відіграють ключову роль у забезпеченні ефективного і надійного обміну даними.

Оверлейні мережі в системах туманних обчислень використовуються для забезпечення зв'язку між різними агентами і вузловими агентами. Ці мережі дозволяють структурувати та організувати комунікацію в середовищі, де різні частини системи можуть бути розподілені фізично. Вони

реалізуються за допомогою спеціалізованих алгоритмів, які визначають маршрутизацію даних та взаємодію між агентами.

Агенти в оверлейній мережі виконують різноманітні завдання, включаючи збір та передачу даних, обробку і аналіз інформації, а також керування ресурсами. Вони можуть функціонувати як самостійні обчислювальні вузли або як частина більшої мережі, спільно вирішуючи завдання відповідно до специфікацій системи.

Вузлові агенти, у свою чергу, відповідають за координацію дій агентів, організацію комунікаційних каналів та забезпечення цілісності даних у системі туманних обчислень. Вони забезпечують маршрутизацію даних, керування навантаженням і оптимізацію ресурсів, щоб забезпечити високу ефективність та надійність роботи системи в цілому.

Цей підхід дозволяє створювати складні мережі, що адаптуються до змінних умов і вимог, забезпечуючи одночасно високу ступінь гнучкості та масштабованості. Оверлейні мережі виявляються необхідним інструментом для оптимізації обчислювальних ресурсів та забезпечення ефективного управління даними у сучасних системах туманних обчислень.

3.3 Визначення показників для перебудови ролей агентів

Для перебудови ролей агентів у системі туманних обчислень важливо встановити оптимальну частоту перевірки необхідності такої перебудови. У розділі 3.1 були визначені умови для додавання нових вузлових агентів на основі інтенсивності вхідних заявок. Однак необхідно також розглянути, як часто потрібно перевіряти необхідність перебудови мережі та її складових елементів.

Розглянемо, як часто потрібно перевіряти необхідність перебудови ролей агентів у системі туманних обчислень залежно від інтенсивності вузлових елементів і їх кількості.

Коли інтенсивність вузлових елементів є низькою і стабільною, можна

вважати, що перевірка необхідності перебудови може проводитись рідше. Наприклад, якщо кількість вхідних заявок на кожний вузловий елемент мінімальна і не змінюється з часом, перевірка може виконуватися раз на мінімальний заданий проміжок часу. Це дозволяє уникнути надмірної витрати ресурсів на перебудову та зберігати стабільність системи.

У випадку помірної інтенсивності вузлових елементів, яка може змінюватися в залежності від різних факторів, таких як час доби чи сезонність, рекомендується здійснювати перевірку частіше.

Це дозволяє більш точно адаптувати систему до змінюваних умов та запобігає накопиченню значного відставання від оптимального режиму роботи.

У випадку високої інтенсивності вузлових елементів, коли кількість вхідних заявок значно змінюється або перевищує встановлені межі, рекомендується проводити перевірку частіше. Наприклад, це може бути кілька разів на день або кожну годину, в залежності від конкретних потреб системи. Це дозволяє оперативно реагувати на зміни навантаження і забезпечує максимальну ефективність роботи системи туманних обчислень.

Отже, перевірка умови перебудови системи можна сформулювати наступним чином:

$$f_{\text{перебудови}} = \begin{cases} \text{досягнення } \Delta t \\ \frac{\lambda_{\text{вихідна}}}{\lambda_{\text{вхідна}}} < 0.5 \end{cases}, \quad (3.6)$$

де: Δt – гранично допустимий час;

$\lambda_{\text{вихідна}}$ – інтенсивність вихідного потоку даних;

$\lambda_{\text{вхідна}}$ – інтенсивність вхідного потоку даних.

3.4 Розрахунок максимальної кількості вузлових агентів

Важливою складовою ефективного функціонування системи туманних

обчислень є динамічне регулювання кількості вузлових агентів, що виконують основну частину обчислень та керують трафіком даних. Для забезпечення оптимальної роботи системи, максимальна кількість таких агентів повинна встановлюватись динамічно залежно від поточного навантаження та умов роботи. Однак слід враховувати, що ця кількість не повинна перевищувати кількість звичайних агентів у системі, які виконують менш ресурсоємні завдання. Встановлення обмеження на кількість вузлових агентів порівняно з кількістю звичайних агентів дозволяє уникнути перевантаження системи, забезпечує рівномірний розподіл навантаження та підвищує загальну стабільність.

В попередніх підрозділах вже було розглянуто динамічного регулювання кількості вузлових агентів, спершу визначаються початкові параметри системи, такі як інтенсивність вхідних заявок та вихідних потоків даних. Початкова кількість вузлових агентів встановлюється на основі поточного навантаження, що дозволяє оптимально розподілити ресурси на початковому етапі роботи системи. Подальше регулювання кількості вузлових агентів здійснюється за допомогою моніторингу змін інтенсивності вхідних та вихідних потоків даних. Якщо інтенсивність вхідних заявок зростає, система автоматично збільшує кількість вузлових агентів, проте ця кількість завжди повинна бути меншою або рівною кількості звичайних агентів. Це забезпечує баланс між різними типами агентів та запобігає надмірному навантаженню на окремі вузли системи.

У цьому ж підрозділі варто розглянути максимальну допустиму кількість вузлових агентів, враховуючи коефіцієнт відносності. Цей коефіцієнт визначається як відношення кількості вузлових агентів до звичайних агентів і не може дорівнювати одиниці. Значення коефіцієнта відносності впливає на точність даних та пропускну здатність мережі. При збільшенні коефіцієнта відносності точність даних, які збираються з агентів, зменшується, однак пропускну здатність мережі зростає.

З метою забезпечення оптимального балансу між точністю даних і

пропускною здатністю мережі, важливо встановити граничні значення для коефіцієнта відносності. Наприклад, для конкретної мережі можна визначити, що коефіцієнт відносності не повинен перевищувати певного значення, яке залежить від технічних характеристик системи та вимог до точності даних.

Таким чином, коефіцієнт відносності елементів визначається наступною формулою:

$$k_{\text{відносності}} = \frac{N_{\text{вузлових}}}{N_{\text{звичайних}}}, k_{\text{відносності}} < 1, \quad (3.6)$$

де: $N_{\text{вузлових}}$ – кількість вузлових агентів у системі;

$N_{\text{звичайних}}$ – кількість звичайних агентів у системі.

Це обмеження дозволяє керувати мережею таким чином, щоб підтримувати необхідний рівень точності даних, при цьому забезпечуючи достатню пропускну здатність для обробки інформації.

Варто зазначити, що коефіцієнт відносності не може бути більше одиниці, оскільки в такому випадку кількість вузлових агентів перевищуватиме кількість звичайних агентів. Це призведе до того, що деякі вузлові агенти будуть простоювати без залучення, що є неефективним використанням ресурсів системи.

Таким чином, встановлення коефіцієнта відносності меншим за одиницю дозволяє забезпечити оптимальне співвідношення між вузловими та звичайними агентами, що сприяє ефективному функціонуванню мережі. Це співвідношення також забезпечує високу точність даних, зібраних звичайними агентами, та ефективне використання пропускної здатності мережі.

Тому для досягнення максимальної ефективності та точності даних важливо ретельно контролювати коефіцієнт відносності і дотримуватися обмеження, що він має бути меншим за одиницю.

Динамічне регулювання також передбачає зменшення кількості вузлових агентів у разі зниження навантаження, що дозволяє економити

ресурси та підтримувати ефективність системи на високому рівні. Таким чином, динамічне встановлення максимальної кількості вузлових агентів, з урахуванням обмеження їхньої кількості відносно звичайних агентів, є важливим аспектом, який забезпечує стабільну та ефективну роботу системи туманних обчислень.

4 МОДЕЛЬНИЙ ЕКСПЕРИМЕНТ

У попередніх розділах було детально розглянуто теоретичні аспекти та основні концепції створення та функціонування системи вузлових і звичайних агентів. Були проведені дослідження важливих параметрів, таких як коефіцієнт відносності, що впливає на точність даних і пропускну здатність мережі, та встановили оптимальні межі для забезпечення ефективної роботи системи.

Моделювання є ключовим етапом у перевірці теоретичних концепцій та припущень. За допомогою моделювання буде зможега побачити, як система функціонуватиме в реальних умовах, і виявити можливі недоліки або області для покращення. Цей процес включає створення віртуальної моделі мережі, налаштування параметрів вузлових і звичайних агентів, а також аналіз результатів, отриманих під час симуляції.

Основна мета цього розділу – оцінити ефективність запропонованої системи, зокрема визначити, як різні значення коефіцієнта відносності впливають на роботу мережі. Ми також розглянемо, як зміна кількості вузлових і звичайних агентів впливає на загальну продуктивність та точність зібраних даних. На основі отриманих результатів буде зроблено висновки щодо доцільності використання певних конфігурацій та надано рекомендації для подальшого вдосконалення системи.

Таким чином, у цьому розділі буде проведено практичне моделювання роботи системи, що дозволить отримати цінні інсайти та підтвердити, або спростувати теоретичні передбачення. Це стане важливим кроком на шляху до створення ефективної та надійної системи агентів, здатної задовольнити вимоги до точності даних та пропускну здатності мережі.

Всього передбачено 4 етапи тестування з різними параметрами. Для тестування системи було розроблено програму на мові програмування C# для модуляції поведінки. Також програма логувала звіти і на основі цих звітів були

побудовані графіки, які допоможуть наглядніше побачити залежності моделі та актуальність запропонованого методу перебудови.

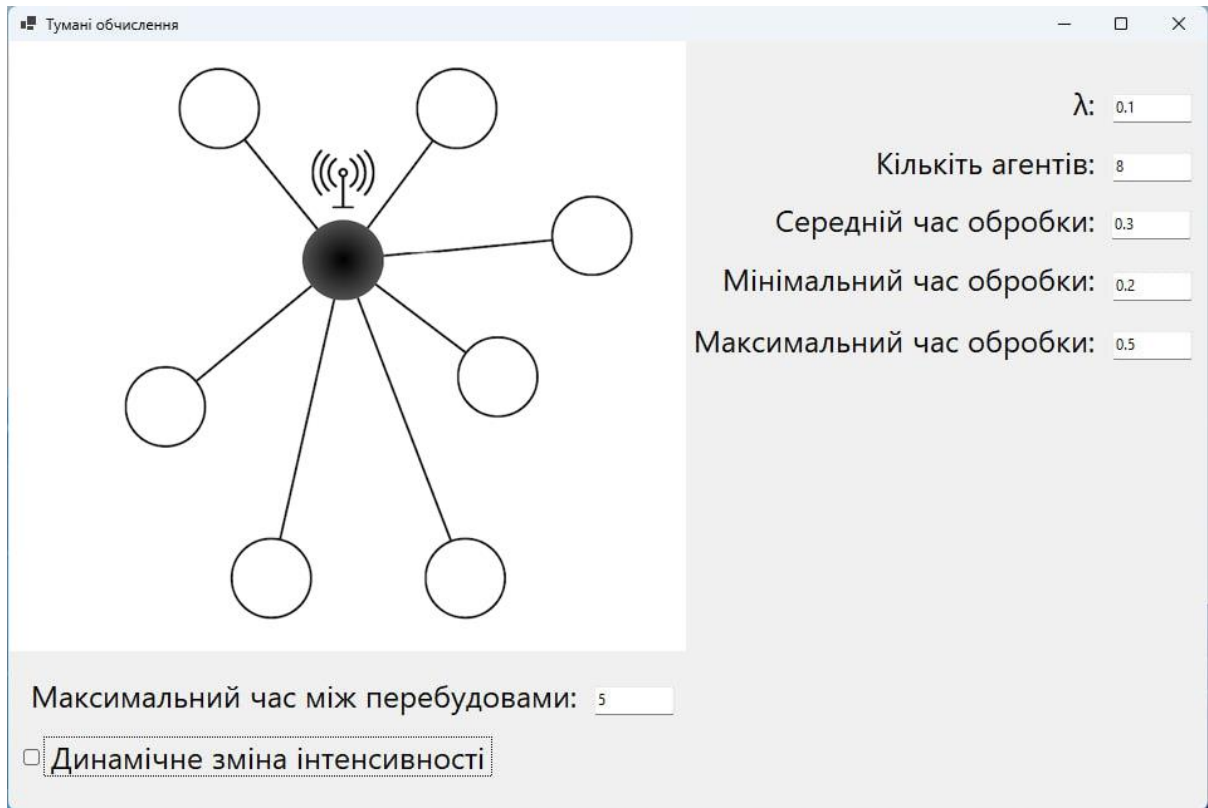


Рисунок 4.1 – Тестування з незмінною вхідною інтенсивністю

При проведенні тестування зі сталою інтенсивністю не відбувається перебудова графу, бо система впорається з зазначеним навантаженням. Це свідчить про стабільність та ефективність обраної архітектури в умовах постійного навантаження. Система демонструє здатність стабільно обробляти вхідні дані без необхідності перебудови своєї внутрішньої структури, що є важливим показником її надійності та витривалості.

Динамічна зміна інтенсивності навантаження дозволяє перевірити адаптивність системи до змінних умов. У випадку значного підвищення інтенсивності може спостерігатись перебудова графу, що забезпечить оптимізацію ресурсів та підтримання належної продуктивності. Це дозволяє системі залишатись ефективною навіть у випадках пікових навантажень, забезпечуючи безперебійну роботу та високу якість обслуговування.

Спробуємо додати динамічну зміну інтенсивності. Зміна графу відображена на рис. 4.2.

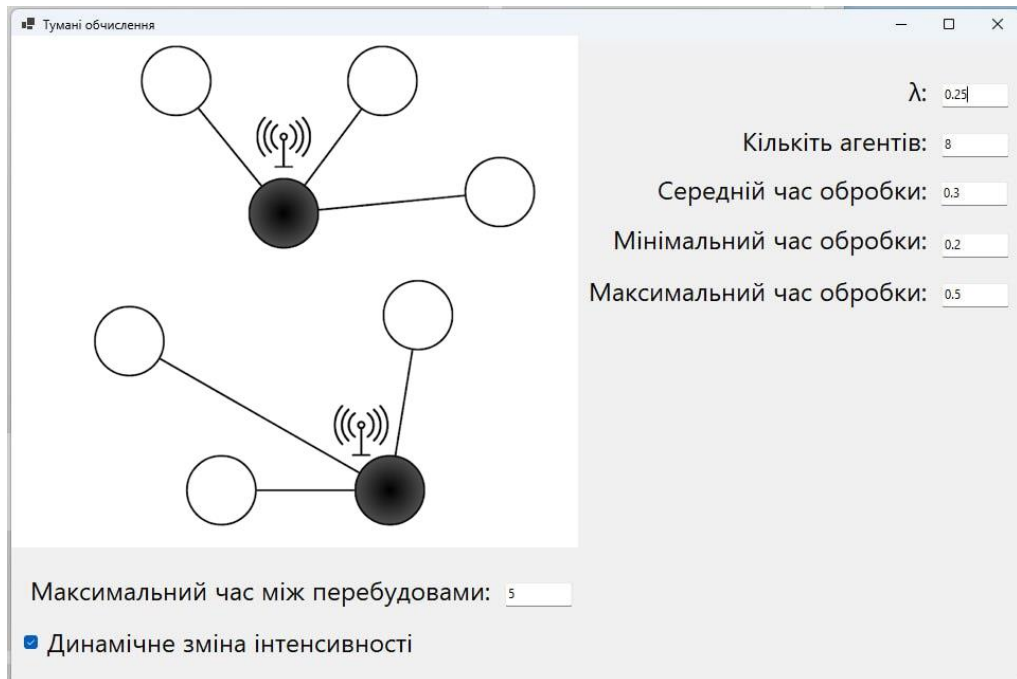


Рисунок 4.2 – Тестування з змінною вхідною інтенсивністю

Отже, можна побачити, що з часом моделювання змінюється кількість вузлових агентів. Зміна навантаження в залежності від часу зображена на графіку 4.1. Варто зазначити, що інтенсивність була збільшена для наглядності.

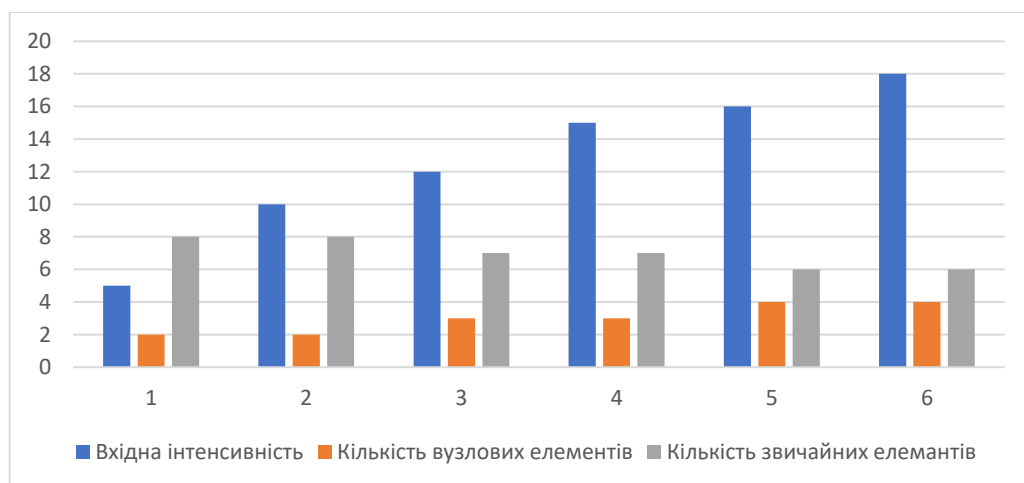


Рисунок 4.3 – Залежність кількості вузлових елементів від інтенсивності

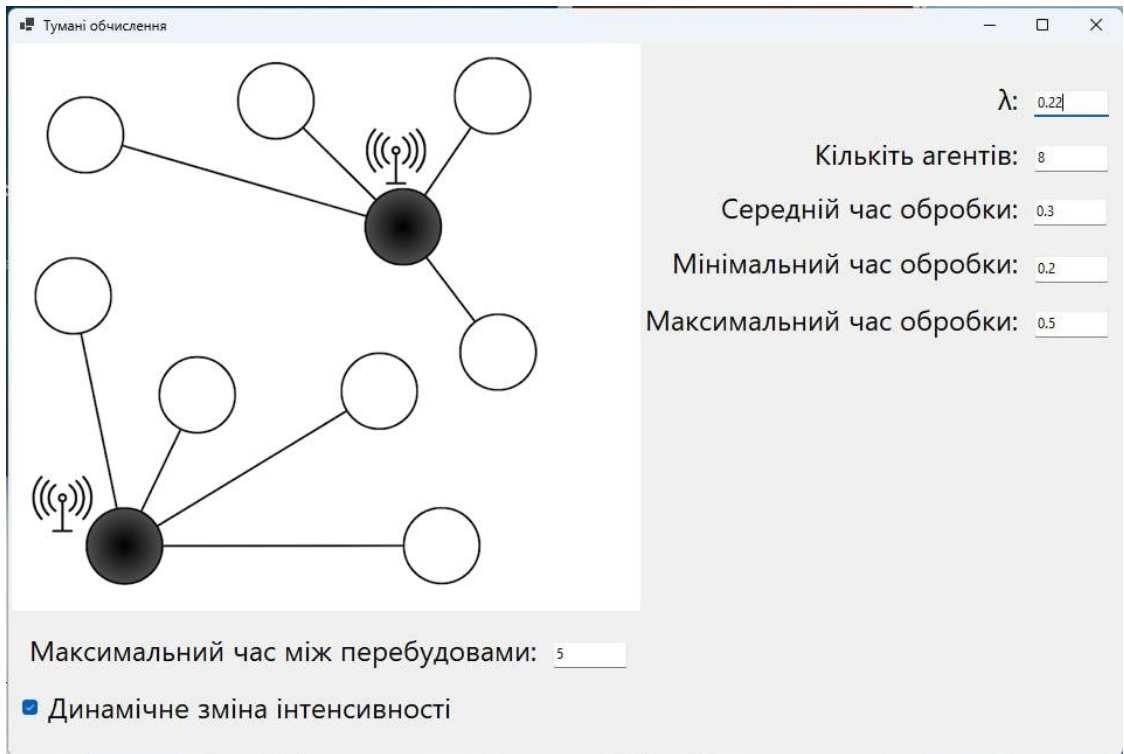


Рисунок 4.4 – Тестування з змінною вхідною інтенсивністю для 10 агентів

Випадок на рисунку 4.4 доволі схожий з випадком зображеним в 4.1. Також для цього випадку графік зображений на рисунку 4.3.

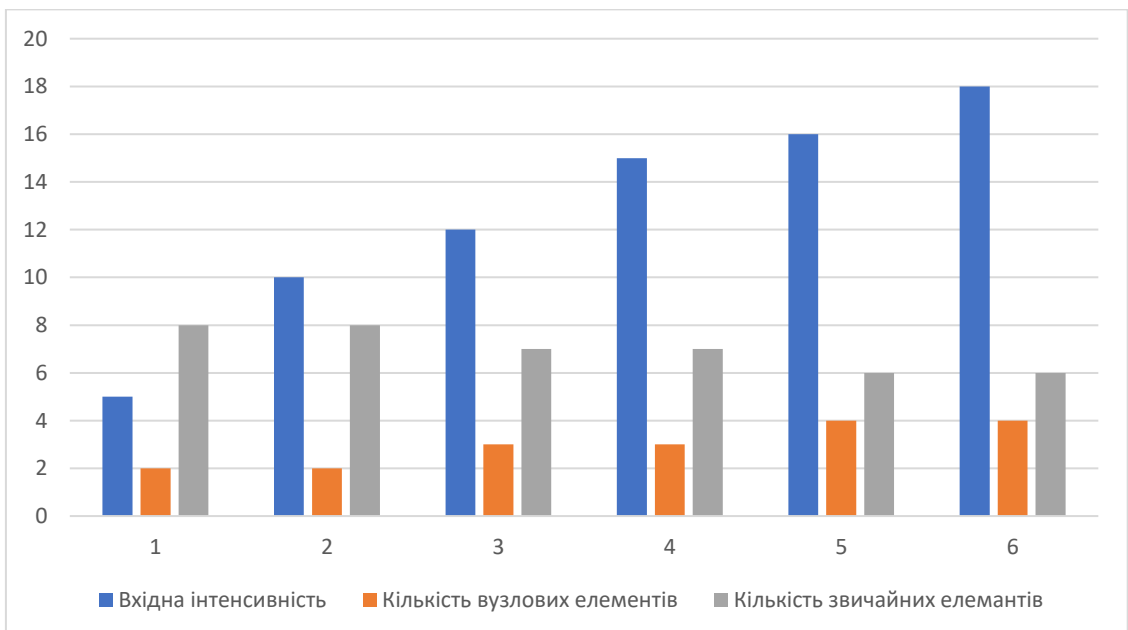


Рисунок 4.5 – Кількості вузлових елементів від змінної інтенсивності

Динамічна зміна інтенсивності навантаження дозволяє перевірити адаптивність системи до змінних умов. У випадку значного підвищення інтенсивності може спостерігатись перебудова графу, що забезпечить оптимізацію ресурсів та підтримання належної продуктивності. Це дозволяє системі залишатись ефективною навіть у випадках пікових навантажень, забезпечуючи безперебійну роботу та високу якість обслуговування.

Такий підхід до тестування дозволяє виявити потенційні слабкі місця в системі та дає змогу внести необхідні корективи для покращення її роботи. Таким чином, використання динамічної зміни інтенсивності є важливим інструментом для забезпечення високої надійності та продуктивності системи в реальних умовах експлуатації.

Наостанок, необхідно провести дослідження з високою вхідною інтенсивністю, щоб остаточно оцінити граничні можливості системи та її здатність до ефективного функціонування при максимальних навантаженнях.

На рисунку 4.6 зображений граф при високій інтенсивності.

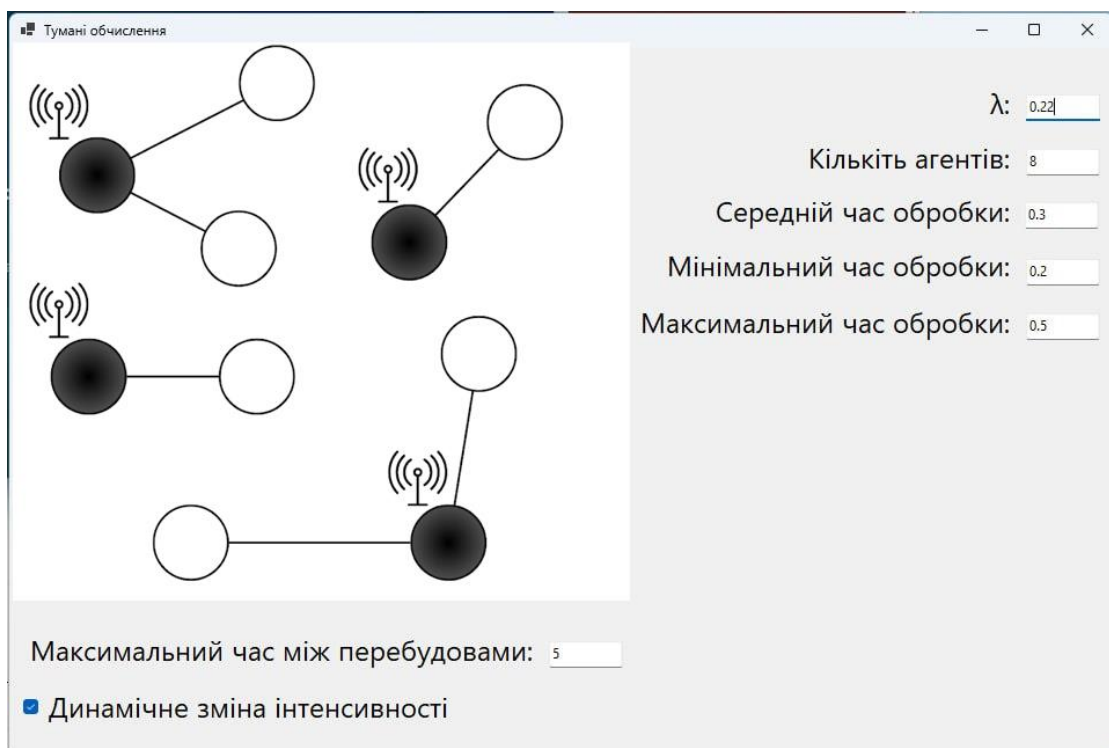


Рисунок 4.6 – Тестування з високою вхідною інтенсивністю

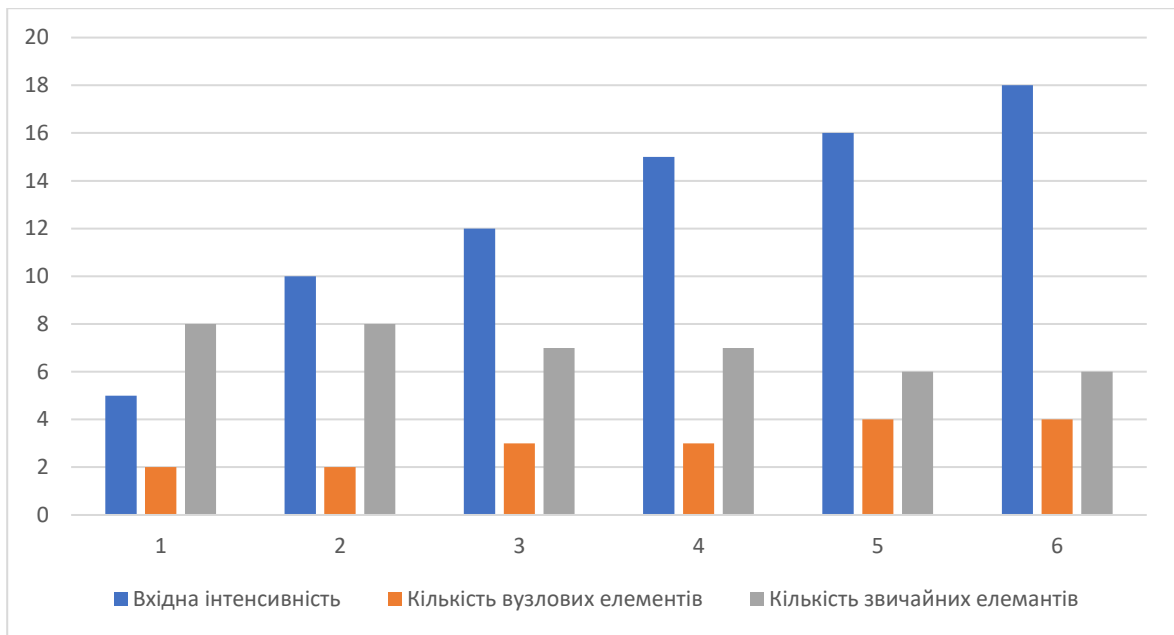


Рисунок 4.7 –Кількості вузлових елементів з високою інтенсивністю

Проведені тести підтверджують, що система працює як очікується, демонструючи стабільність та адаптивність до змінних умов навантаження, забезпечуючи обробку даних і використання ресурсів.

ВИСНОВКИ

У даній роботі було розглянуто актуальні тенденції у розвитку обчислювальних мереж та туманних обчислень. Встановлено, що туманні обчислення є потужним інструментом для моделювання систем з нечіткими даними, що знайшли широке застосування у штучному інтелекті, розпізнаванні образів і прийнятті рішень.

Особлива увага була приділена динамічним обчислювальним оверлейним мережам та їхній ролі у підвищенні ефективності використання розподілених ресурсів та забезпеченні безпеки обчислювальних систем. Висвітлено важливість полігрових мереж як ключового компонента в таких системах.

У рамках цього дослідження було проведено математичні обрахунки і побудовані графіки для оцінки ефективності запропонованої методології туманних обчислень в оверлейних мережах на базі полігрових мереж. Математичні моделі дозволили оцінити час обробки даних та ресурси, які необхідні для їхньої реалізації.

Графіки, побудовані на основі отриманих даних, підтвердили теоретичні передбачення щодо покращення якості обробки даних і зниження часу обчислень в порівнянні з існуючими підходами. Вони також продемонстрували переваги використання полігрових мереж у контексті туманних обчислень, зокрема, їхню здатність до ефективного розподілу завдань та координації ресурсів між вузлами мережі.

Ці результати підкріплюють висновок про актуальність запропонованої методології і підтверджують її потенціал для подальшого застосування у вдосконаленні обчислювальних систем, які можуть адаптуватися до змінних умов роботи. Результати дослідження відкривають нові перспективи для розвитку гнучких і стійких технологій туманних обчислень та оверлейних мереж у сучасних інформаційних технологіях.

Для подальшого вивчення та розвитку запропонованої системи туманних обчислень в оверлейних мережах на базі полігрових мереж можна розглянути її застосування в конкретних прикладних ситуаціях, наприклад, в моніторингу зволоженості лісу за для запобігання пожеж.

Система може бути використана для збору та обробки даних з сенсорів, розташованих у лісовому масиві. За допомогою туманних обчислень, вона зможе адаптуватися до змінних умов та нечітких даних про зволоженість ґрунту і атмосферні умови. Полігрові мережі допоможуть ефективно керувати обміном даних між різними датчиками та центральним обчислювальним вузлом, забезпечуючи оптимальне використання ресурсів і забезпечуючи надійну передачу даних.

Такий підхід дозволить покращити точність та швидкість моніторингу зволоженості лісу, що є критично важливим для вчених, лісовиків та охоронців природи для ефективного управління ресурсами та захисту екосистеми.

Таким чином, подальше дослідження може спрямовуватися на практичні застосування технології в реальних умовах, що дозволить підтвердити її ефективність та потенціал у вирішенні конкретних завдань моніторингу та управління навколишнім середовищем.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Глибинна інтеграція хмарних та туманих обчислень : тези / М.А. Гунько, А. І. Кардаш, В. М. Ткачов : Вісник Херсонського національного технічного університету 4(87), 2023. – 252 с.
2. Alessandro Bassi and Geir Horn. Internet of Things in 2020: A Roadmap for the Future. European Commission: Information Society and Media, 2020.
3. Till Michael Beck, Martin Werner, Sebastian Feld, and Thomas Schimper. Mobile Edge Computing: A Taxonomy. In 6th International Conference on Advances in Future Internet, AFIN'14, pages 48–54, Lisbon, Portugal, November 2014. International Academy, Research, and Industry Association.
4. M. Hunko, V. Tkachov, A. Kovalenko, and H. Kuchuk, “Advantages of fog computing: A comparative analysis with cloud computing for enhanced edge computing capabilities,” in 2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek), pp. 1–5, IEEE, 2023.
5. Mike J. Walker, Betsy Burton, and Michele Cantara. Gartner Hype Cycle 2016. <https://www.gartner.com/doc/3383817>, 2016. Accessed: 2016-09-20.
6. Sukhpal Singh and Inderveer Chana. QoS-Aware Autonomic Resource Management in Cloud Computing: A Systematic Review. ACM Computing Surveys, 48(3):42:1– 42:46, December 2015.
7. Tom H. Luan, Longxiang Gao, Zhi Li, Yang Xiang, and Limin Sun. Fog Computing: Focusing on Mobile Users at the Edge. Computing Research Repository, abs/1502.01815(1), March 2015.
8. Jaeho Kim and Jang-Won Lee. OpenIoT: An Open Service Framework for the Internet of Things. In 2014 IEEE World Forum on Internet of Things, WF-IoT'14, pages 89–93, Seoul, Korea, March 2016. IEEE.
9. Amir Vahid Dastjerdi and Rajkumar Buyya. Fog Computing: Helping the Internet of Things Realize Its Potential. Computer, 49(8):112–116, August 2018.
10. V Tkachov Cellular Technology Based Overlay Networks for the Secure

Control of Intelligent Mobile Objects: Models and Numerical Study/ V Tkachov, A Kovalenko, V Kharchenko, M Hunko // ICTERI 2021: Information and Communication Technologies in Education, Research, and Industrial Applications. – Kherson, Ukraine, October 2, 2021. – Pp. 42-63.

11. V Tkachov Cellular Technology Based Overlay Networks for the Secure Control of Intelligent Mobile Objects/ V Tkachov, A Kovalenko, V Kharchenko, M Hunko // ICTERI 2021: Information and Communication Technologies in Education, Research, and Industrial Applications. – Kherson, Ukraine, October 2, 2021. – Pp. 480-490.

12. M Hunko. Application Architecture For Obtaining Data From Scientometric Databases / M Hunko, V Tkachov, O Liashenko, J Rabčan // 2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek). – Kharkiv, 2022.

13. Tkachov V. Principles of Constructing an Overlay Network Based on Cellular Communication Systems for Secure Control of Intelligent Mobile Objects / Vitalii Tkachov, Andriy Kovalenko, Mykhailo Hunko and Kateryna Hvozdetska // Информационные технологии и безопасность. Материалы XIX Международной научно-практической конференции ИТБ-2020. – К.: ООО "Инжиниринг", 2020. – С. 51-55.

14. Hunko M. A. The software tool for identifying the cellular network status of a mobile device under the Android operating system / M. A. Hunko, K. A. Voropaeva // Радіоелектроніка та молодь у XXI столітті : матеріали 25-го Міжнародн. молодіжн. форуму, 20-22 квітня 2021 р. – Харків : ХНУРЕ, 2021. – Т. 5, секція 4. – С. 102-103.

15. Hunko, M., Ruban, I., Hvozdetska, K.: Securing the Internet of Things via VPN technology. *Comput. Inf. Syst. Technol.* (2021).

16. Tkachov, V. M., and D. V. Hrechmak. "The advancements and impacts of cloud computing: abstraction, innovation, and sustainability." (2023).

17. Vitalii Tkachov, Mykhailo Hunko, Olga Morozova, Artem Tetskyi, Andrii Nicheporuk "Method to Determine Fault-Tolerant Performance Probability of High-Survivability Computer Network based on Mobile Platform"//2021 IEEE

8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T).

18. M Hunko, V Tkachov, A Kovalenko " Advancements in edge and fog computing in limited Internet connectivity and real-time data requirements" - Проблеми інформатизації : одинадцята міжнародна науково-технічна конференція – Харків 2023 – с.41.

19. Гунько М. А. Розробка моделі інтелектуальної мобільної системи для своєчасного запобігання механічних перешкод / М. А. Гунько // "Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 49)" : матеріали Міжнар. наук. Інтернет-конф., 10 червня 2020 р. – Тернопіль, 2020. – С. 7–8.