

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)

Моделі управління хмарними та туманними
обчисленнями в розподілених інформаційних системах

(тема)

здобувач 2 року навчання,

групи _____ СІМ-23-3

_____ **Артем ФУРМАНОВ**

(власне ім'я, прізвище)

Спеціальність _____

_____ 123 «Комп'ютерна інженерія»

(код і повна назва спеціальності)

Тип програми _____ освітньо-наукова

(освітньо-професійна або освітньо-наукова)

Освітня програма _____

_____ Системне програмування

(повна назва освітньої програми)

Керівник: _____ проф. Максим ВОЛК

(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри ЕОМ _____

(підпис)

_____ **Андрій КОВАЛЕНКО**

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

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

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

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Фурманову Артему Андрійович _____
(прізвище, ім'я, по батькові)

1. Тема роботи Моделі управління хмарними та туманними обчисленнями в розподілених інформаційних системах

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

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

3. Вхідні дані до роботи _____

1. Технології моделювання хмарних систем

2. Туманні обчислення (Fog Computing)

3. Існуючі системи управління туманними обчисленнями

4. Технології організації розподілених інформаційних систем

4. Перелік питань, що потрібно опрацювати в роботі _____

1 Аналіз предметної області

2 Моделі управління хмарними та туманними обчисленнями в розподілених інформаційних системах

3 Експериментальні дослідження

4 Висновки

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

Слайд-презентація – 12 слайдів _____

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

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

КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання 21 квітня 2025 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф. Максим ВОЛК
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 56 с., 12 рис., 4 табл, 1 дод., 49 джерел.

ХМАРНІ ОБЧИСЛЕННЯ, ТУМАННІ ОБЧИСЛЕННЯ,
МОДЕЛЮВАННЯ, ІНТЕРНЕТ РЕЧЕЙ, БАЛАНСУВАННЯ
НАВАНТАЖЕННЯ, ЯКІСТЬ ОБСЛУГОВУВАННЯ.

У роботі розглядається проблема управління перевантаженнями в розподілених інформаційних системах на основі хмарних та туманних обчислень. Запропоновано систему динамічного управління перевантаженнями, яка забезпечує ефективний розподіл обчислювальних ресурсів і покращує якість обслуговування (QoS) в середовищі Інтернету речей (IoT). Основна ідея системи полягає у використанні механізмів пріоритетного обслуговування запитів та адаптивного балансування навантаження. Проведені експерименти з використанням симуляторів iFogSim та CloudSim підтвердили зменшення часу обробки запитів та підвищення ефективності використання ресурсів. Отримані результати демонструють перспективність впровадження запропонованої системи для оптимізації роботи IoT-інфраструктури.

ABSTRACT

Master's thesis: 56 pages, 12 figures, 4 tables, 1 appendice, 49 sources.

CLOUD COMPUTING, FOG COMPUTING, MODELING, INTERNET OF THINGS, LOAD BALANCING, QUALITY OF SERVICE.

The paper considers the problem of congestion management in distributed information systems based on cloud and fog computing. A dynamic congestion management system (DCMB) is proposed, which provides effective allocation of computing resources and improves the quality of service (QoS) in the Internet of Things (IoT) environment. The main idea of the system is to use mechanisms for priority request service and adaptive load balancing. Experiments using the iFogSim and CloudSim simulators confirmed the reduction of request processing time and increased resource efficiency. The results obtained demonstrate the prospects of implementing the proposed system for optimizing the operation of the IoT infrastructure.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ.....	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Хмарні обчислення	11
1.2 Сучасні особливості Інтернету речей (IoT).....	14
1.3 Fog Computing	15
1.4 Питання захисту даних у хмарних та туманних обчисленнях	18
1.5 Постановка мети та завдань дослідження	20
2 МОДЕЛІ УПРАВЛІННЯ ХМАРНИМИ ТА ТУМАННИМИ ОБЧИСЛЕННЯМИ В РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ	22
2.1 Сучасні моделі управління хмарними та туманними обчисленнями....	22
2.2 Модель управління хмарними та туманними обчисленнями в розподілених інформаційних системах	27
2.2.1 Користувачі хмари.....	29
2.2.2 Брокер хмарних послуг	29
2.2.3 Брокер туманних послуг	31
2.2.4 Постачальник хмарних послуг	32
2.3 Висновки по другому розділу.....	33
3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ	34
3.1 Опис методу організації експериментів	34
3.2 Опис моделювання.....	36
3.3 Опис результатів моделювання	39
3.4 Висновки по третьому розділу	41

	7
ВИСНОВКИ.....	42
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	45
ДОДАТОК А.....	50

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

ВМ – віртуальна машина
ПЗ – програмне забезпечення
ЦП – Центральний процесор
API – Application Programming Interfase
CoT – Cloud of Things
DCM – Dynamic Congestion Management
DIS – Distributed Interactive Simulation
DEVS – Discrete Event System Specification
IaaS – Infrastructure as a Service
IoT – Internet of Things
FaaS – Fog as a Service
FCFS – First-Come First-Served
FIFO – First In, First Out
FPGA – Field-Programmable Gate Array
FSB – Fog Service Broker
GKE – Google Kubernetes Engine
HFD – Host Fault Detection
HPF – Highest Priority First
HTTP – Hypertext Transfer Protocol
PaaS – Platform as a Service
SaaS – Simulation as a Service
SLA – Service-Level Agreement
VM – Virtual Machine
QoS – Quality of Service
WFQ – Weighted Fair Queuing

ВСТУП

Хмарні обчислення забезпечують величезну кількість ресурсів, які динамічно розповсюджуються на вимогу клієнтів через Інтернет. Типові централізовані моделі хмарних обчислень можуть мати труднощі з вирішенням проблем, спричинених додатками Інтернету речей, такими як збій мережі, обмеження потужності, затримка тощо.

Інтернет речей (IoT), відповідно до рекомендації ІТУ ІТУ-Т У.2060, був визначений як: «глобальна інфраструктура для інформаційного суспільства, яка надає передові послуги та взаємозв'язок (фізичних і віртуальних) речей на основі існуючих засобів обробки сумісної інформації та технологій зв'язку». Серед фізичних об'єктів можна виділити прибори та різного роду сенсори та датчики. Як обчислювальні вузли – віртуальні машини. Канали зв'язку – телекомунікаційні мережі.

Існує кілька аспектів сучасної діяльності Інтернету речей, які дозволяють подальшу автоматизацію та оптимізацію. Основними перевагами є дружність до людини та адаптація до виконання конкретних завдань, покращення використання енергії, забезпечення безпеки, управління через автоматизацію та використання штучного інтелекту.

Одним із запроваджених методів вирішення цих проблем є туманні обчислення, які роблять хмару ближчою до пристроїв IoT. Посередницька система динамічного управління перевантаженнями досліджується у даній кваліфікаційній роботі. За допомогою цієї запропонованої системи забезпечуються вимоги до якості обслуговування (QoS) IoT, які визначаються угодою про рівень обслуговування (SLA).

Основною ідеєю є забезпечення виконання великої кількості запитів, що надходять із шару посередника туману. Крім того, запроваджено політику переадресації, яка допомагає хмарному посереднику послуг у виборі та пересиланні високопріоритетних запитів до відповідних хмарних ресурсів від

брокерів туману та користувачів хмар. На цю запропоновану ідею впливає зважена справедлива черга (WFQ). В дослідженні використовувався механізм черги CISCO для спрощення керування та контролю перевантажень, які можуть мати місце на стороні брокера хмарних служб. Система, запропонована в цій роботі, оцінено за допомогою інструментів iFogSim і CloudSim, і результати демонструють, що він покращує відповідності вимогам IoT (QoS), а також сприяє уникненню порушень хмарних SLA.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Хмарні обчислення

Хмарні обчислення можна розглядати як масивний динамічний і розподілений ресурс для зберігання даних, надання комунікаційних послуг та обчислювальних потужностей, доступних за запитом користувачам через Інтернет [1,2].

Традиційно ресурси хмарних обчислень розташовуються у величезному центрі зберігання даних і керуються та контролюються третьою стороною, яка пропонує обчислювальну інфраструктуру щоб користувачі хмари могли використовувати по принципу "будь-хто з будь-якого місця через Інтернет" [3]. Хмарні обчислення представляють собою величезний ресурс, до якого можна отримати доступ за запитом і розподілити виконання завдань через Інтернет [4–8]. Рисунок 1.1 ілюструє середовище моделі хмарних обчислень. Хмарні служби дозволяють користувачам (зокрема фізичним особам, підприємствам і клієнтам) використовувати програмне забезпечення та апаратне забезпечення, яким у віддалених місцях керують треті сторони. Такі хмарні приклади включають веб-пошту, онлайн-сховище файлів і сайти соціальних мереж [9].

Хмарні обчислення швидко стали дуже популярною та широко використовуваною мережевою системою через свою вартість, ефективність, зниження витрат користувача та підвищення прибутковості для хмарних провайдерів [10–12]. Системи хмарних обчислень можуть отримувати доступ до комп'ютерних ресурсів і даних з будь-якого місця Інтернет, підключених до мережі [13].

Запитувачі послуг хмари можуть платити постачальникам за хмарні послуги на основі певних типів стягнення плати, такі як оплата за користування, такі ж, як комунальні послуги (наприклад, телефон, газ,

електроенергія та вода), за допомогою яких клієнти можуть стягувати щомісячну плату або щоквартально на основі використання ними фіксованих одиниць [14,15,16].

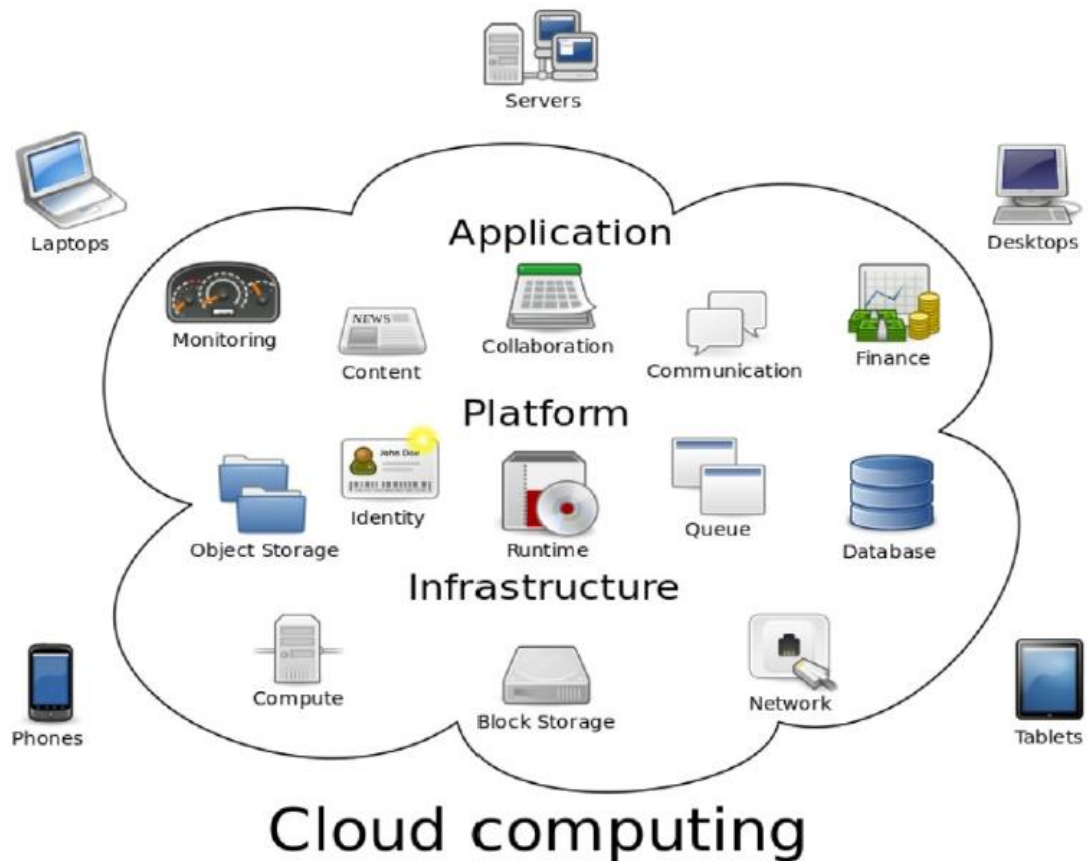


Рисунок 1.1 – Середовище хмарних обчислень

На сьогодні застосування хмарних обчислень значно розширилося, що робить їх найважливішою частиною майбутнього покоління доступних обчислювальних інфраструктур та послуг. Їх вигідні характеристики дозволять користувачам хмари використовувати та розвивати свої ресурси на основі плати за використання, як показано на рисунку 1.2 [17]. Хмарні сервіси мають багато моделей, таких як платформа як послуга (PaaS), інфраструктура як послуга (IaaS) і програмне забезпечення як послуга (SaaS), які пропонуються постачальниками хмари, як описано нижче (рисунок 1.2). Можливе комбіноване використання моделей різних рівнів з забезпеченням важливих властивостей як гнучкість, надійність, масштабування та інш.

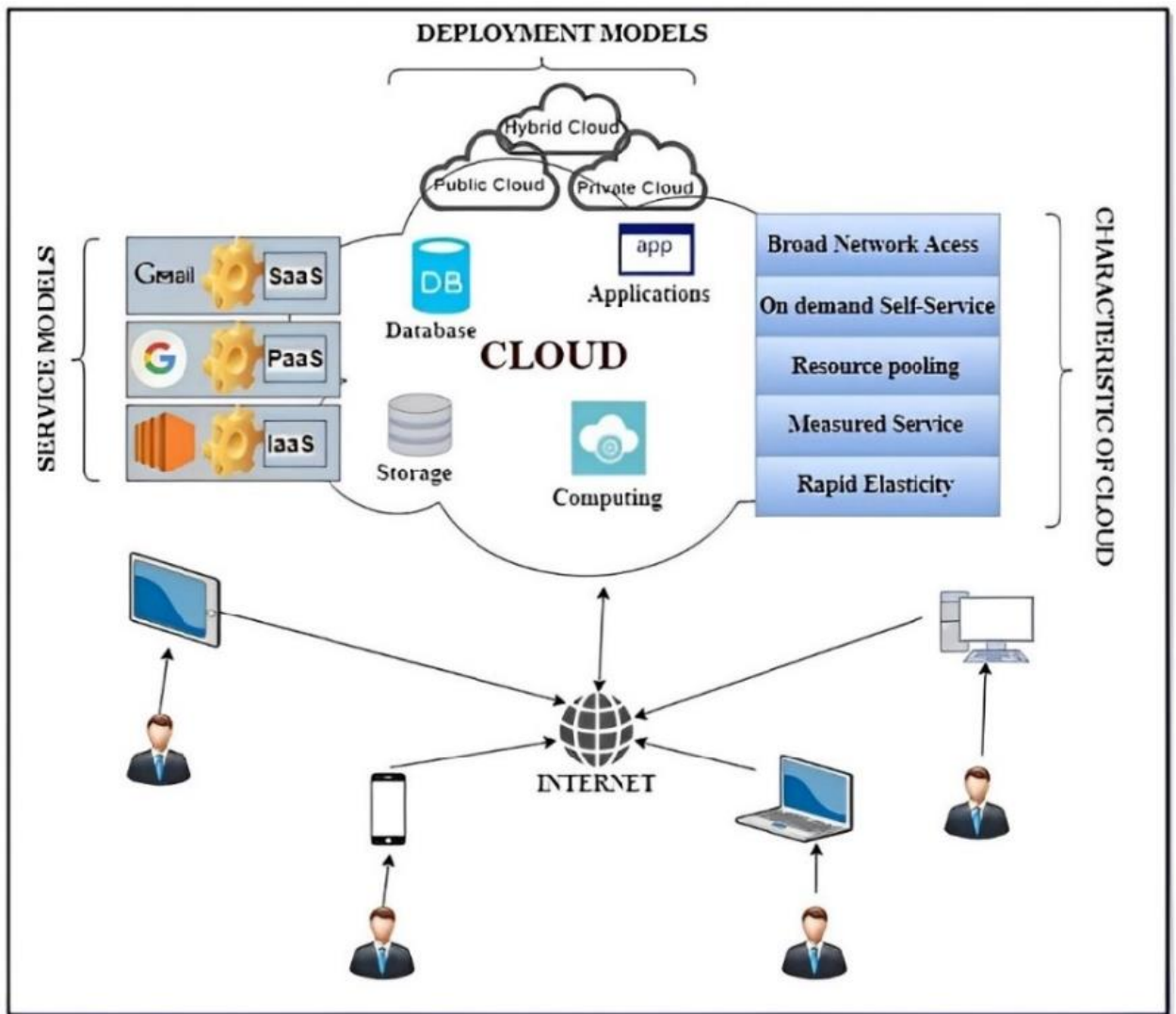


Рисунок 1.2 – Моделі послуг хмарних обчислень

IaaS пропонує ресурси віртуалізації за потреби (на вимогу) як частину сервісної бізнес-моделі [18–20], де постачальник є власником інструментів та обладнання, чиє використання він пропонує хмарним користувачам віртуально замість того, щоб їх купувати (за схемою оплата за використання) [14]. Для керування операціями (графічний інтерфейс користувача) зазвичай використовується Веб-консоль. Крім того, ця модель сервісу дозволяє користувачам хмари впроваджувати самообслуговування після використання послуг провайдера. До всіх інструментів організується доступ через комп'ютерні мережі.

Хмарний сервіс моделі SaaS є найпоширенішим порівняно з іншими моделями. Завдяки йому користувач може використовувати та отримувати

доступ до програм без необхідності купувати або завантажити їх. Провайдер надає клієнту потужності в оренду [18–21]. Крім того, хмарні користувачі можуть використовувати служби SaaS і отримувати до них доступ через Інтернет браузер.

PaaS дозволяє користувачам хмари орендувати апаратну та програмну інфраструктуру з хмари провайдерів, які використовують Інтернет для запуску своїх програм [18,19,22]. PaaS в основному використовується розробниками програм для створення та тестування своїх розробок.

1.2 Сучасні особливості Інтернету речей (IoT)

IoT стосується збільшення кількості фізичних об'єктів, які пов'язані засобами Інтернет. Є багато прикладів цих об'єктів, які включають системи керування та моніторингу, такі як кондиціонування повітря, опалення та вентиляція в smart будинках. IoT дозволяє фізичним об'єктам чути, думати та бачити, а також створювати завдання, які дозволяють щоб вони могли спілкуватися один з одним щодо узгодження рішень та інформаційного обміну. IoT перетворює ці об'єкти з традиційних форм на розумні об'єкти за допомогою використання додаткових технологій. Приклади цих розумних об'єктів універсальні та поширюються на обчислення, комунікаційні технології, вбудовані пристрої, сенсорні мережі, програми та Інтернет-протоколи [23]. На рисунку 1.3 показана загальна концептуалізація IoT середовища.

Програми IoT спершу розроблялися для домашніх мереж, розумних будівель, промисловості, автоматизації від інтелектуального керування транспорту до повсюдної охорони здоров'я [24–26]. Наприклад, розумні будинки будуть з використанням пристроїв IoT для виявлення можливих витоків газу, керування температурою навколишнього середовища, якістю повітря, контролю проникнення в будинки та багато інших задач, що стосуються будинків та їх жителів [27].

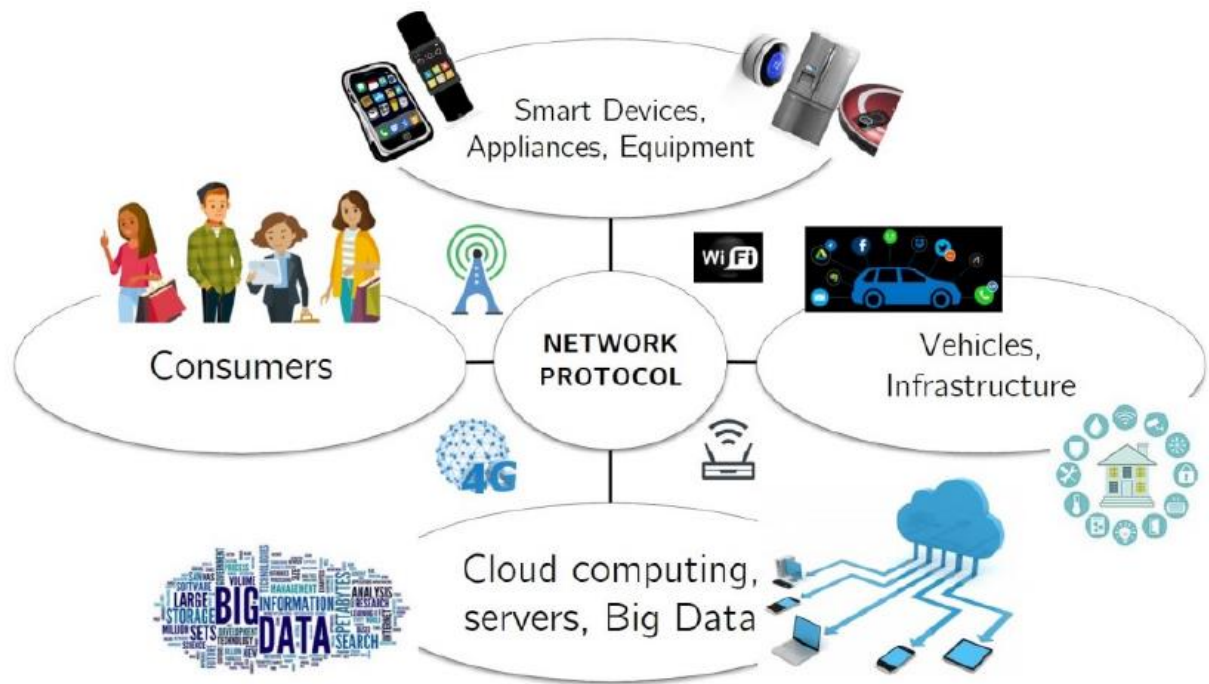


Рисунок 1.3 – Середовище IoT [28]

1.3 Fog Computing

Оскільки IoT відрізняється обмеженою потужністю зберігання та обробки, він також має відносно низьку продуктивність, конфіденційність, надійність та безпеку. Поєднання IoT і хмари за допомогою мереж створює Cloud of Things (CoT), що є одним із найкращих способів подолати майже всі недоліки IoT [28, 29]. Крім того, CoT може оптимізувати процеси збору даних IoT, обробку, інтеграцію та розгортання [2,29]. Створення нових додатків IoT ускладнюється через величезну кількість пристроїв IoT з різномірними парадигмами [30].

Датчики та інші пристрої надають величезні обсяги даних у додатках IoT, генеруючи великі дані, які згодом обробляються та аналізуються, щоб прийняти відповідні дії.

Дані, зібрані датчиками, потрібно аналізувати в хмарі, що вимагає високої пропускної здатності для використовуваної мережі. Таким чином, ці проблеми можна вирішити за допомогою туманного обчислення [30,31].

Кілька сфер, особливо IoT, можуть отримати вигоду від туманних обчислень – нової технології що пропонує багато переваг. Технологія була запропонована фірмою Cisco та забезпечує спосіб обробки даних поблизу джерела інформації, що робить її більш швидкодоступною та ефективною [32].

Користувач IoT може скористатися перевагами туманних обчислень для доступу до таких послуг, як зберігання й обробка даних. Подібно до того, як хмара надає послуги користувачам, туманні обчислення надають можливість для пристроїв Інтернету речей підключатися до цих послуг і використовувати їх. Замість того, щоб передавати дані в хмару, туманні обчислення пропонують можливість локального зберігання та обробки для пристроїв туману. Зберігання, обчислення та мережеві ресурси доступні як через хмарні обчислення, так і через туманні обчислення [33].

Як показано на рисунку 1.4, туманні обчислення розширюють хмару, щоб наблизити середовище до речей, які взаємодіють з даними IoT (наприклад, таких як пристрої користувача).

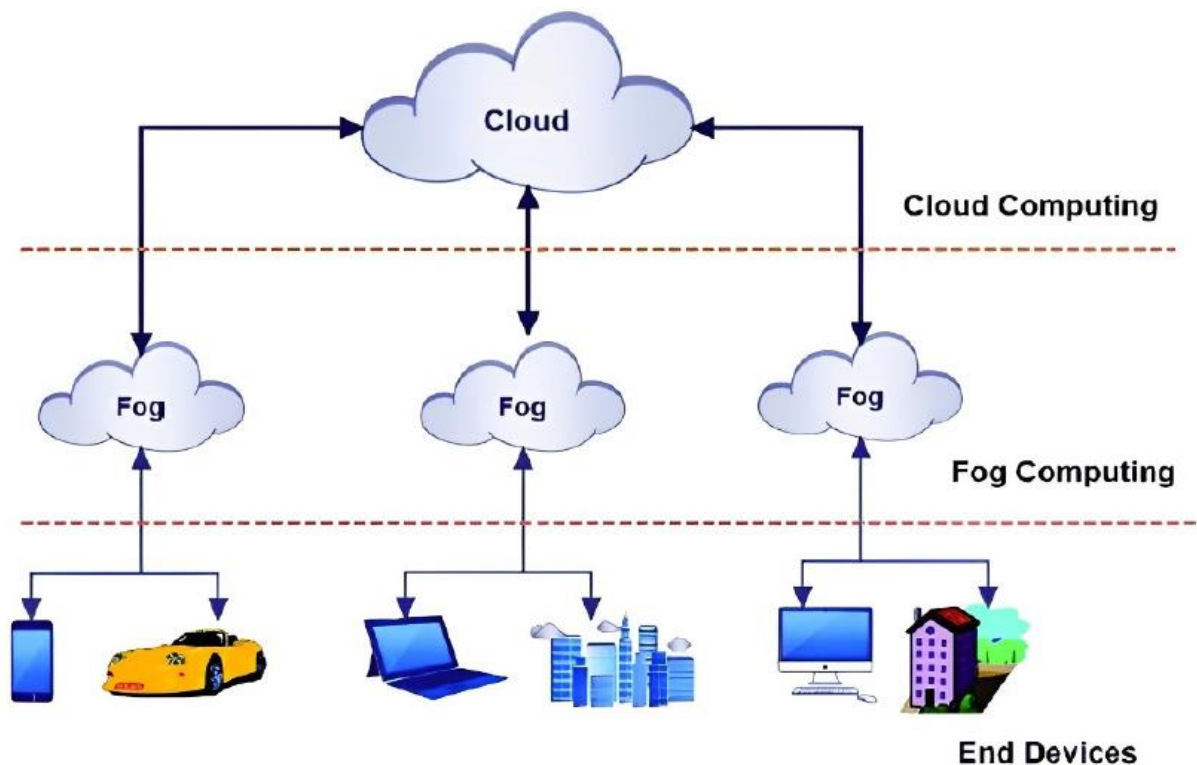


Рисунок 1.4 – Туманні обчислення наближають хмару до кінцевих пристроїв

Туманні обчислення можна розглядати як розширення хмарних обчислень до рівня мережі, де дані обробляються поблизу джерела, а не надсилаються в хмару. З збільшенням кількості датчиків, генерується все більше даних. В минулому, ці дані були б оброблені та збережені в хмарі. Однак цей підхід має декілька недоліки, включаючи затримку та перевантаження мережі [34].

Fog як послуга (FaaS) – це новий клас послуг, що став можливим завдяки Інтернету Речі та туманним обчисленням. Тут постачальники послуг створюють масиви туманних вузлів на розподіленій географічній території, що діють як ресурси, які можуть бути орендовані підприємствами багато різних галузей. Кожен туманний вузол має локальні обчислення, сховище та мережу можливості, що робить його ідеальним рішенням для розподіленої обробки та аналізу даних [35].

FaaS пропонує підприємствам новий спосіб пропонувати послуги клієнтам. На відміну від великих компаній, які зазвичай керують хмарами в різних масштабах, FaaS надають публічні та приватні послуги з обчислень, керування та зберігання для великих і малих підприємств. В результаті вони можуть задовольнити потреби широкого кола клієнтів [36].

Розробники або створюють порти туманних вузлів, або додатки IoT на краю мережі. Туманні вузли, розташовані поблизу краю мережі, поглинають зібрані дані з пристроїв IoT, програма fog IoT спрямовує ці дані в оптимальне для них місце для аналізу. Є три типи даних [37].

Найбільш чутливі до часу дані: туманний вузол аналізує ці типи даних поблизу речей (наприклад, сенсорні системи), які створюють дані.

Дані, які можуть бути відкладені на хвилини або секунди для відповіді або дій: це дані направляються до вузла агрегації для аналізу, оцінки та дії.

Дані, менш чутливі до затримки, пересилаються в хмару для архівування та аналіз, довгострокового зберігання та аналізу великих даних. У нашій роботі ми робимо деякі пропозиції:

- пропонується нова посередницька система динамічного управління

перевантаженнями (DCMB), яка, по суті, запозичена з моделей черги Cisco DCMB та може обслуговувати термінові запити (роботи) з шару туману;

- система може одночасно керувати величезною кількістю хмарних запитувачів беручи до уваги вимоги QoS клієнтів, як це передбачено SLA;

- система DCMB використовується для надання хмарним провайдерам завдань і моніторингу їх прогресу їх обслуговування.

1.4 Питання захисту даних у хмарних та туманних обчисленнях

В останні роки інформаційні технології стрімко змінюються. Хмарні обчислення додали більш перспективну роль ІТ для користувачів. Хмарні обчислення надають постачальникам свої послуги в оренду за погодинними ставками.

Вони також здають в оренду простір користувачам у своїх фізичних системах. Однак ці служби мають кілька загроз безпеці для користувачів. Про це йдеться у звіті Cloud Security Alliance – зловживання, незахищені інтерфейси та зловмисне використання вразливі загрози. Ці загрози були пов'язані з інтерфейсами прикладних програм в хмарному середовищі.

Інформаційна безпека поділяється на три основні цілі, такі як цілісність, конфіденційність і доступність. До загроз безпеці відноситься проблема довгострокової конфіденційності, оскільки враховується, що поточні схеми шифрування не є безпечними. Є вразливість до витоку інформації – це ще одне занепокоєння, оскільки дані передаються на зовнішні служби. Підробка даних також створює загрози для конфіденційності даних.

Оскільки нові технології з'являються для задоволення потреб користувачів, є вимоги значного підвищення безпеки хмарних погроз. Ці загрози виникають у вигляді кількох невидима експлуатаційних модулів через служби хмарних обчислень і пов'язаних з ними інтерфейсами. Це стало істотним викликом для протидіяти виниклим і потенційним атакам.

Присутність незахищених інтерфейсів є великою проблемою для обох хмар користувачів і постачальників хмарних послуг. Головним чином безпека та доступність хмарних сервісів залежить від API, який включає доступ до даних і шифрування даних хмари. Можна провести подальші дослідження, щоб переконатися безпеки цих API та мережевих інтерфейсів. Нові пропозиції можуть вирішити проблеми захисту послуг від навмисних та випадкових нападів та порушення умов послуг. Крім того, багат шарові API мають більшу складність тому, що сторонні оператори використовують хмарні сервіси.

Фактичні власники ресурсів не можуть отримати доступ до послуг. Крім того, шкідливі "інсайдери" є звичайною загрозою для хмарних служб, оскільки вони порушують умови послуг і доступ до інформації, до якої вони мають доступу. Зазвичай працівник – це зловмисний інсайдер, який конфіденційну інформацію, яка належить компанії або її законним користувачам, може пошкодити або скопіювати в однорангових системах обміну файлами.

Multi-Tenancy - технологія спільного використання ресурсів з асоційованими ризиками конфіденційності та цілісності даних. Мульти-оренда була названа серйозною проблемою для професіоналів в хмарних обчисленнях. Розуміння професіоналів про атаку і вектора атаки є найбільш важливим. Це призвело до збільшення кількості користувачів послуг хмарних обчислень, які піклуються про загрозу безпеці даних і конфіденційності.

Користувачі хмари стикаються з проблемами надійності з боку великих провайдерів хмарних обчислень. Amazon's EC2 і S3 зазнали збою на 3 години 8 годин у лютому 2008 р. та липні 2008 р. відповідно. Gmail від Google послуги залишалися недоступними протягом 3 годин, що завадило 113 мільйонів користувачів доступу до документів через їх облікові записи Gmail. Microsoft і Amazon запропонували своїм клієнтам доступ до вибору геопросторового розташування компаній центрів обробки даних.

Користувачі хмарних служб, як клієнти, так і зловмисники, повинні мати права автентифікації користувачів для доступу до хмарних служб.

Спілкування між користувачами та хмарними ресурсами прагне захищеного каналу, який зберігає інформацію для входу користувачів безпечніше, ніж для користувачів протоколу передачі гіпертексту (HTTP). По-друге, спілкування між хмарними сервісами та користувачами має бути забезпечена синхронізацією.

1.5 Постановка мети та завдань дослідження

Сформулюємо основні проблеми та недоліки сучасних рішень, які усуває наша робота та принципи їх усунення.

По-перше, це обмежена здатність до масштабування: адаптивний алгоритм балансування навантаження підтримує ефективну інтеграцію нових пристроїв у систему.

По-друге, надмірне навантаження на центральні обчислювальні вузли – DCMB знижує навантаження завдяки розподіленому управлінню.

І останньою проблемою є висока затримка обробки запитів – застосування механізму пріоритетного обслуговування дозволяє прискорити виконання критично важливих завдань.

Метою роботи є підвищення ефективності управління ресурсами в гетерогенних IoT-системах шляхом оптимізації розподілу навантаження, скорочення часу обробки запитів та покращення якості обслуговування (QoS) через впровадження динамічної системи керування перевантаженнями (DCMB).

Для реалізації мети в роботі необхідно вирішити наступні задачі:

- провести аналіз існуючих підходів до управління ресурсами в хмарних та туманних обчисленнях;
- розробити алгоритм пріоритетного обслуговування та адаптивного балансування навантаження;

- впровадження запропонованої системи у середовище симуляції (iFogSim, CloudSim).
- оцінка ефективності системи на основі експериментальних результатів.

Об'єктом роботи є процеси управління хмарними та туманними обчисленнями в розподілених інформаційних системах.

Предмет дослідження: моделі та алгоритми управління ресурсами, синхронізацією даних та комунікацією у розподілених IoT-системах.

У розділі 2 розглядаються моделі хмари та туману і представлено запропоновану модель, яка експериментально оцінюється в розділі 3.

2 МОДЕЛІ УПРАВЛІННЯ ХМАРНИМИ ТА ТУМАННИМИ ОБЧИСЛЕННЯМИ В РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

2.1 Сучасні моделі управління хмарними та туманними обчисленнями

У цьому розділі розглядаються найпопулярніші моделі для хмарних і туманних обчислень, механізми моніторингу та надання ресурсів, а також описується CSB робочих місць. Завдяки цій інформації можна краще зрозуміти сильні та слабкі сторони кожній з цих моделей і вибрати ту, яка найкраще відповідає сформульованим потребам.

Система DRPM [38] – мультиагентна система, яку було розроблено для контролю ресурсів хмарного провайдера з урахуванням вимог клієнтів до якості обслуговування, які контролюються SLA. У події фізичного перевантаження машини, алгоритм виявлення помилок хоста (HFD) системи DRPM визначає віртуальні машини, щоб пом'якшити проблему. Це досягається врахуванням джерела перевантаження та прийняття обґрунтованого рішення по перерозподіл ресурсів. Систему DRPM можна оцінити та протестувати інструментом CloudSim, щоб проаналізувати ресурси, зменшити енергоспоживання та уникнути порушень SLA.

Система Aneka [39] забезпечує PaaS для хмарних середовищ на платформі додатків на основі NET. Ця система дозволяє легко керувати, розгортати та розробляти програми в хмарі. API та середовище виконання застосунків, представлених цією системою, можна використовувати в загальнодоступній і приватній хмарі, таких як GoGrid і Amazon EC2. Крім того, модель Aneka пропонує методи моніторингу ресурсів відповідно до вказівок SLA, які допомагають переконатися, що нові завдання від користувачів хмари виконуються вчасно та в межах параметрів, встановлених SLA. Це робиться шляхом оцінки часу, необхідного для виконання цих завдань та зіставлення їх із наявними ресурсами. Система

продовжує працювати, якщо розраховано час завершення для нових робочих навантажень; інакше вона продовжує працювати з хмарними ресурсами та залишатися в рамках SLA.

Туманні обчислення були описані в [33] разом із його застосуваннями в реальному часі, і було показано, що туманні обчислення можуть запускати та обробляти великі дані, створені IoT-пристроями. Крім того, було продемонстровано, що туманні обчислення можуть усунути затримку та проблеми із заторами. Цей метод працює шляхом оцінки часу, необхідного для завершення нового завдання від користувачів хмари, а потім зіставлення їх із доступними ресурсами та термінами виконання SLA.

В роботі [36] показано, що IoT і туманні обчислення є двома важливими новими технологіями, які починають бачити більше інтеграції. Вони надають уяву про деякі проблеми, пов'язані з розвитком систем Інтернету речей, і про те, як туманні обчислення можуть допомогти вирішити їх. Крім того, вони обговорили, як розвивати нові можливості для бізнесу завдяки використанню сучасних архітектур зберігання, обчислення та мереж. На додаток до огляду характеристик і переваг цієї архітектури туману, автори також рекомендували рішення для кількох проблем IoT.

Є статті, які розглядають інші архітектури туманних обчислень, такі як [40]. Замість того, щоб використовувати хмару, їх модель використовує IoT у місцевому тумані, або служби туману розташовані в межах програмно-визначеного ресурсу рівня управління [41,42]. Використовуючи проміжне програмне забезпечення на основі хмари, вузли туману аналізуються, упорядковуються, і забезпечуються необхідними засобами.

В [43] описана архітектура CSB, яка показана на рисунку 2.1, де архітектура CSB розділена на чотири частини.

Інтерфейс користувача між користувачем і CSB служить сполучною ланкою між ними. Інтерпретатор додатків описує, як мають виконуватися завдання, який QoS має бути надано, і те, що має бути реалізовано користувачем. Крім того, службовий дані необхідні для виконання, що

визначаються інтерпретатором. Кілька інших основних послуг включені в ці вимоги, наприклад, послуги визначення місця розташування послуги. Щоб отримати доступ до необхідних послуг, Інтерпретатор облікових даних перевіряє облікові дані.

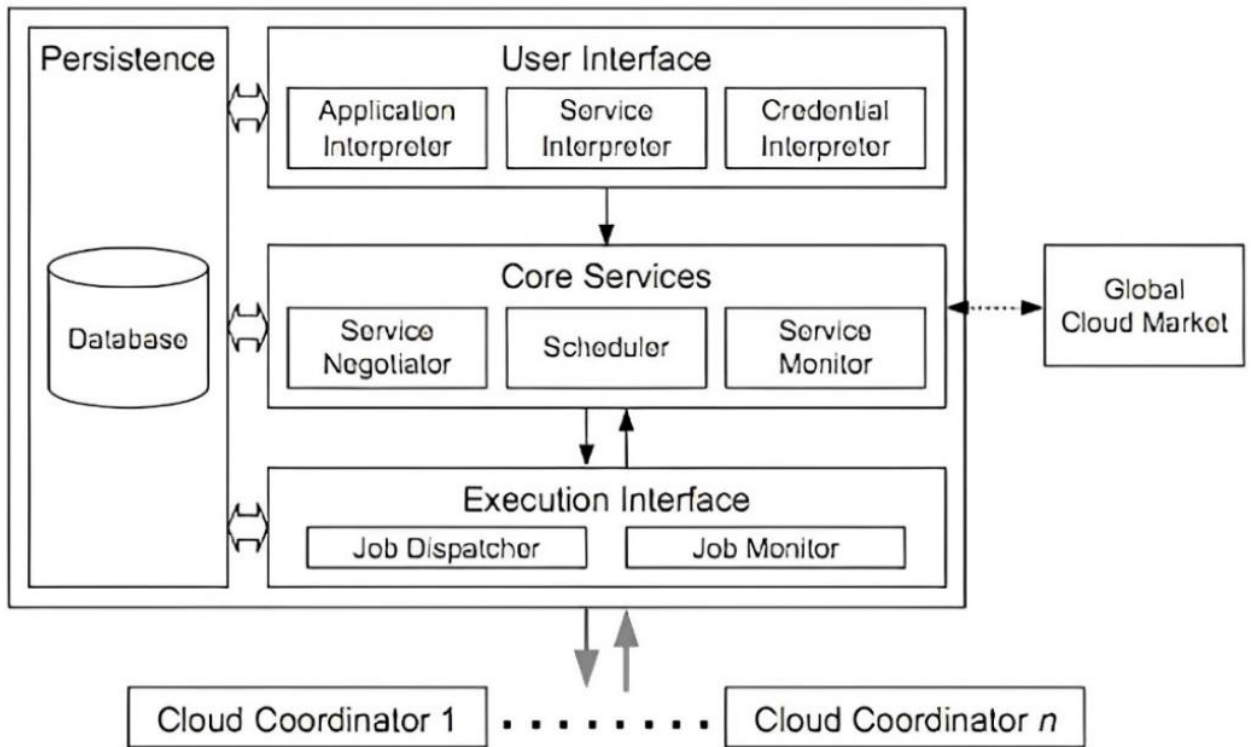


Рисунок 2.1 – Архітектура брокера Хмарних сервісів

Рівень основних служб використовується CSB для відповідності запитам користувачів із найбільшою кількістю відповідних хмарних сервісів. Сервісний інтерфейс збирає вимоги від користувача і передає їх планувальнику, який потім використовує ці вимоги для визначення, які хмарні служби найкраще задовольняти потреби користувача. Сервісний монітор – це постійно запущена програма для перевірки доступності хмарних служб і пошуку доступних нові послуги. Це дозволяє відстежувати стан хмари та гарантує, що сервіси є завжди в робочому стані.

Інтерфейс виконання – це модуль, де диспетчер завдань виконує програми користувача поєднання файлів даних із кодом програми користувача та надсилання отриманих пакетів до хмарних ресурсів. Цей

рівень забезпечує необхідну інфраструктуру. Монітор завдань є важливим інструментом для відстеження прогресу роботи та забезпечення того, що результати передаються замовнику після завершення.

Рівень стійкості є ключовим для підтримки стану інтерфейсу користувача, інтерфейсу виконання і основних послуги в разі відмови брокера.

В [44] представили нову систему під назвою Dynamic Congestion Management (DCM), який використовує кілька агентів для розгляду різних елементів під час прийняття рішень, наприклад, як обмеження хмарних провайдерів, кількість і типи доступних ресурсів, які є важливими факторами в прийнятті рішень, для виконання вимог клієнтів до QoS. На рисунку 2.2 показано динамічне керування перевантаженнями в (DCM) архітектурі Система розділена на три частини.

Кожному клієнту надається локальний агент, який відповідає на запити у порядку пріоритету. Крім того, агент несе відповідальність за відправку запитів до класифікатора всередині CSB.

У разі перевантаженості роботи класифікатор Cloud Service Broker визначає пріоритет завдань, які будуть виконуватися в першу чергу. Коли користувачі хмари продовжують надсилати запити про роботу до CSB у великій кількості, CSB може випадково відхилити деякі з них. Класифікатор у системі DCM прагне розгорнути апріорні категоризації завдань локального агента для керування запитами, що надходять до CSB, щоб контролювати випадкові відхилення запитів. Якщо CSB не перевантажений, класифікатор реалізує порядок FIFO для вхідних запитів. І навпаки, може бути застосована зважена справедлива черга (WFQ) для усунення перевантаження [45].

Фізичні та віртуальні машини постачальника послуг надають визначені характеристики, включаючи пропускну здатність, завантаження ЦП, обсяги сховища даних, величину оперативної пам'яті тощо, та отримують вимоги до них під час роботи від CSB через агентів для кожного окремого ресурсу, який надсилає до Job Monitor з періодичним оновленням повідомлень.

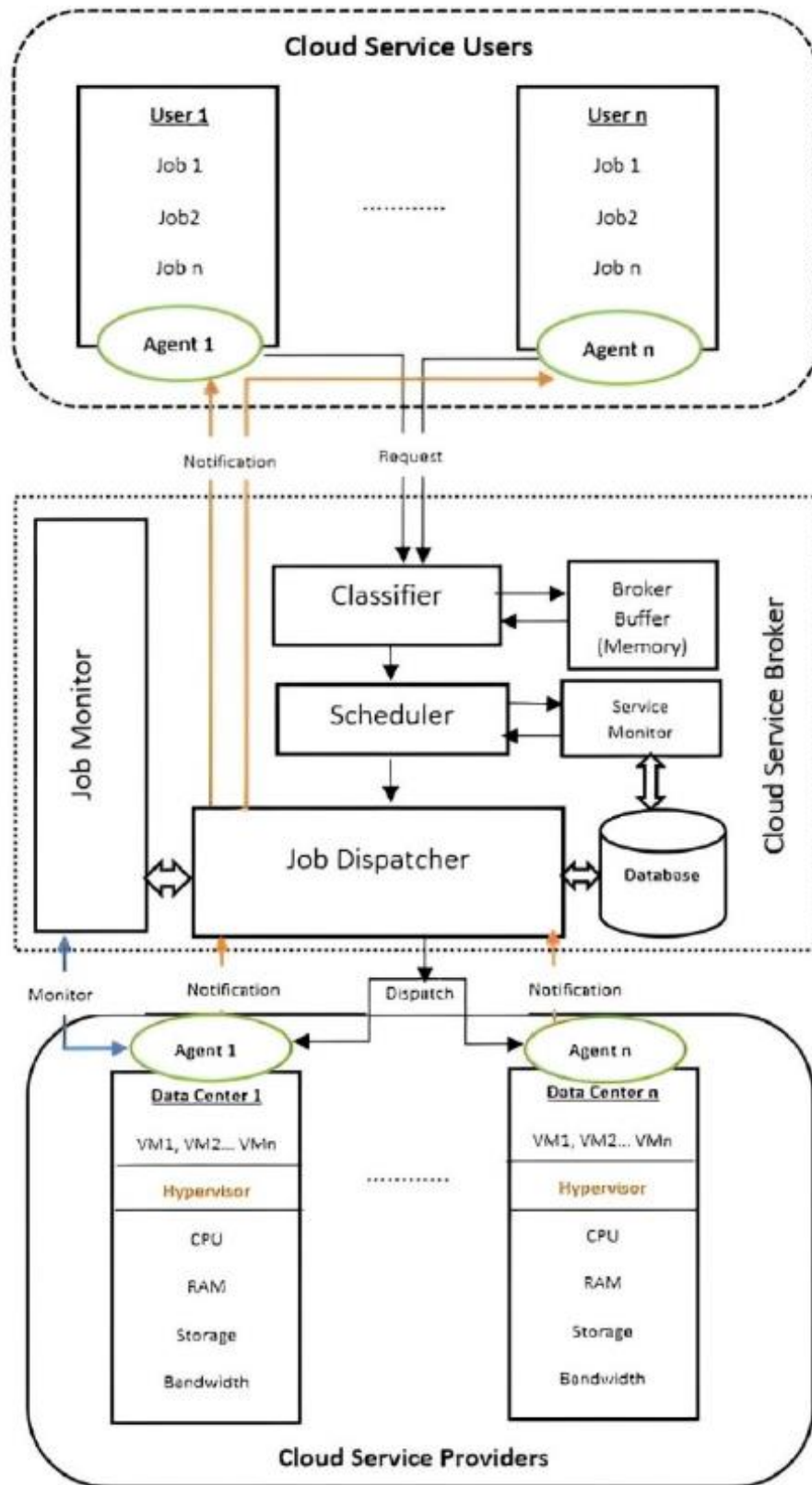


Рисунок 2.2 – Архітектура системи динамічного керування перевантаженнями (DCM)

Через обмежені ресурси (пам'ять, енергія тощо) обчислення спільних даних може бути трудомістким і зазвичай не може виконуватися самими

пристроями IoT. Компанії змогли подолати це обмеження потужності, передавши обчислення на аутсорсинг процесів у хмару завдяки прийняттю та використанню хмарної парадигми. Однак, це розвантаження призводить до втрати якості наданого обслуговування (QoS), включаючи підвищення затримки, викликаних відстанню між хмарою та кінцевими пристроями, зростання накладних витрат мережі та підвищений ризик безпеки та конфіденційності [46]. За останні кілька років обчислювальна парадигма пропагується як спосіб зменшити ці недоліки. Граничні обчислення дозволяють передавати обчислювальні завдання на вузли, розташовані ближче до кінцевих пристроїв (за один крок від них). Як наслідок, ці обов'язки ближче до джерела даних і споживача даних, що призводить до підвищення якості послуг.

Однак периферійні обчислення, на думку деяких дослідників, спрямовані на розвантаження обчислювальної роботи від хмари до вузлів туману, у той час як інші стверджують це лише сукупність пристроїв на боці комп'ютерного ланцюга, включаючи IoT прилади тощо. Вкрай важливо приділити увагу технологіям, які дозволять працювати в реальному часі по збору та аналізу даних, а також забезпечують якість процесу та продукції [47].

Таким чином, необхідно запровадити засоби обчислень, які гарантують якість продукції процес і кінцеві продукти (бездефектне виробництво) [48].

Крім того, деякі дослідження радять використовувати периферійні обчислення для різних доменів, застосувань із жорстким часом реакції, а також для тих, хто бажає знизити вартість своєї інфраструктури або краще контролювати свою конфіденційність.

2.2 Модель управління хмарними та туманними обчисленнями в розподілених інформаційних системах

Пропонована багатоагентна система DCMB розроблена з урахуванням

кількох факторів під час прийняття рішень з використанням різних агентів для прийняття обґрунтованих рішень.

Ця система була створена з метою покращення умов дорожнього руху шляхом зменшення заторів та покращення транспортного потоку. Деякі фактори, які враховуються при у даному рішенні включають термінові запити (роботи), що надходять із шару туману, великої кількості хмарних запитувань, технічні вимоги клієнтів і QoS, а також обсяг і обмеження ресурсів хмарних провайдерів. Ця система по суті запозичена з черги Cisco та системи DCM.

На рисунку 2.3 показано архітектуру системи DCMB, яка складається з чотирьох основних модулів: Користувачі хмари (Cloud User), Брокер хмарних послуг (Cloud Service Broker), Брокер послуг Туману і Постачальник хмарних послуг (Cloud Service Provider).

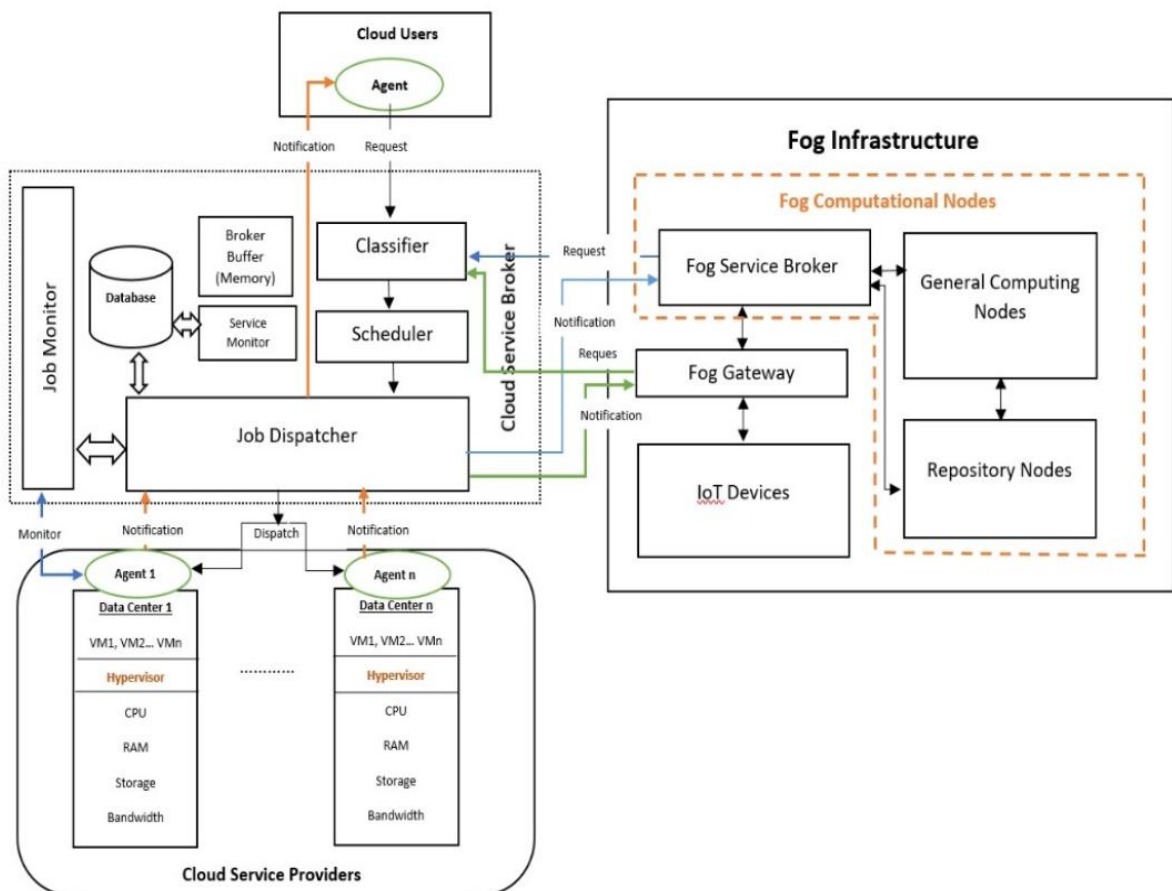


Рисунок 2.3 – Архітектура системи динамічного управління перевантаженнями

2.2.1 Користувачі хмари

Цей модуль призначає агента кожному клієнту, який отримує запити клієнтів відповідно до їх пріоритету та пересилає виділені запити до CSB (зокрема до свого класифікатора). Маркування, яке виконує місцевий агент, вважається додатковою послугою, за допомогою якої клієнти можуть організувати пріоритетне використання хмари. Якщо ця послуга не обрана, рівень пріоритету буде призначено місцевим агентом. Запити користувачів містять умови роботи (специфікації), в тому числі в частині специфікацій обладнання, обов'язкове програмне забезпечення та тип віртуальної машини (VM). Місцевий агент присвоює запитам користувачів низький, середній або високий пріоритет, що дозволяє користувачеві виконувати деякі складні завдання коли потрібно.

2.2.2 Брокер хмарних послуг

Брокер хмарних послуг курує хмарним центром. Користувачі хмари (включаючи пристрої Інтернету речей і брокер туману) надсилають завдання до CSB, яка є ядром системи DCMB. Під час заторів, класифікатор визначає, яку роботу слід виконувати першою. Завдяки великій кількості запитів, які отримує CSB, він може бути змушений відкласти або відхилити деякі запити. Замість того, щоб затримати або відхилити запити згідно з чергою FIFO, класифікатор може використовувати запит маркування, попередньо застосований агентом як у частині користувача, так і в брокері туману на рівні туманних обчислень, для класифікації запитів таким чином, щоб відповідати вимогам QoS клієнтів.

Алгоритм FIFO використовується класифікатором для обробки вхідних запитів, якщо немає перевантаження в CSB. У разі перевантаження класифікатор покладається на WFQ для вирішення задачі вибору наступного запиту. Потoki автоматично сортуються в черги за допомогою WFQ, який

призначає вагу до кожної черги відповідно до її пріоритету. Спочатку процес йде по колу, де фрагменти часу розподіляються для кожного процесу в рівних відсотках, але після завершення раунду, потік трафіку пропускається процесором залежно від його ваги. Механізм WFQ вирішує проблему голодування, яка може виникнути, коли використовується циклічна система.

Рисунок 2.4 пояснює роботу класифікатора під час отримання запитів користувачів хмари, запитів пристроїв IoT та брокерів, використовуючи механізм WFQ.

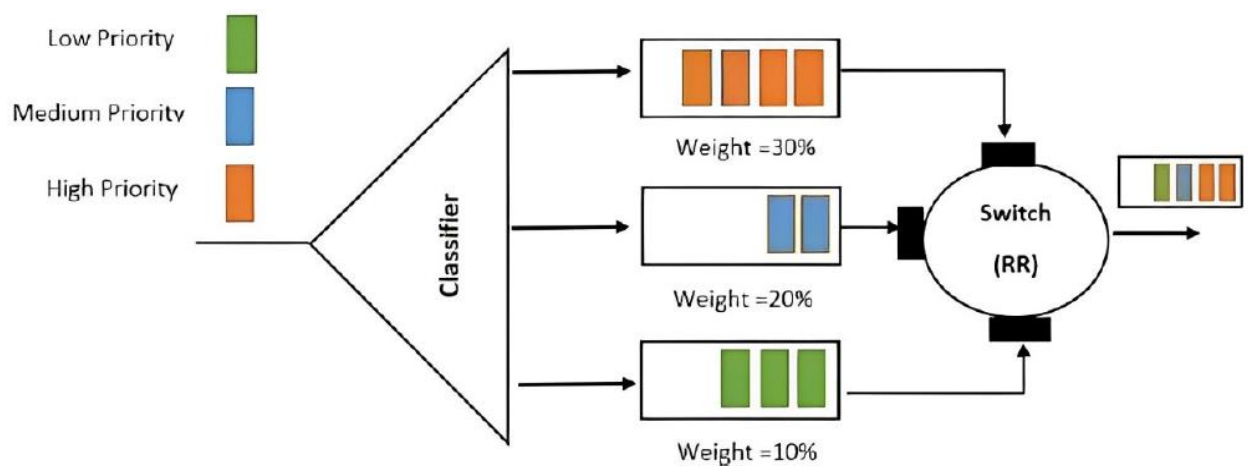


Рисунок 2.4 – Механізм класифікатора

На наступному кроці планувальник запускає запити, які класифікатор вибрав для виконання першими. Відповідаючи на запит користувача, планувальник вибирає найбільш підходящий хмарний сервіс на основі характеристик заявки та вимог. Статус хмарної служби контролюється модулем Service Monitor і нові хмарні служби періодично перевіряються. Згодом диспетчер завдань отримує завдання користувачів і об'єднує програму користувача та файли даних, які надсилаються до певного хмарного ресурсу, для їх виконання. Стан виконання роботи контролюється монітором завдань, щоб знову надіслати результати завдання клієнту як тільки роботи будуть завершені. Статус завдання можна переглядати та керувати ним за допомогою агента, який пов'язаний разом із хмарним

ресурсом. У випадках, коли брокер зазнає невдачі (наприклад, система брокера не працює або стався збій), стан кожного об'єкта CSB збережено в базі даних для відновлення після усунення несправності. Нарешті, коли завдання клієнту завершено, агент, підключений до ресурсу, спрямовує результати назад до джерела завдання.

2.2.3 Брокер туманних послуг

У модулі Брокеру туманних послуг реалізується високий рівень інтеграції IoT з edge-fog та встановлюються їх відносини з хмарою. Деякі компоненти в цій частині адаптовані з моделі, що було описано вище. Пристрої IoT спостерігають за зовнішнім середовищем і перетворюють будь-яку дану директиву у фізичні дії.

Пристрої IoT підключаються до найближчих шлюзів за допомогою дротового або бездротового зв'язку за мережевими протоколами. Fog Gateway реалізує користувацькі інтерфейси для програм, щоб полегшити користувачам увід облікових даних, повідомлення про очікування служби, доступ до серверної програми, отримання результатів служби, запит ресурсів з обчислювальної інфраструктури відповідно до доступності та керування ними у взаємодії з пристроями IoT. Він використовує або протокол обмеженого застосування (CoAP), або простий мережний протокол керування (SNMP) для зв'язку. Загальні обчислювальні вузли відповідають за зберігання даних, виконання програм та інші функції операційного управління.

Вузли сховища зберігають інфраструктуру та облікові дані користувача, а також дані з пристроїв IoT; створюють резервну копію каталогу програм; збільшують хмарне сховище для допомоги в операціях керування даними.

Посередник служби туману (FSB) відповідає за роботу та керування шаром туману операції, в тому числі:

- пошук відповідних обчислювальних вузлів для запуску найбільш чутливих до часу даних, які потребують швидкої реакції шару туману для запитів, які не можуть чекати відгуку від хмари;

- пошук відповідного обчислювального вузла(ів) або відповідного вузла сховища для запуску або зберігання даних, які можуть бути відкладені на хвилини або секунди для відповіді або дій;

- надсилання менш чутливих до часу даних у хмару із затримкою для архівування та історичного аналізу, довгострокове зберігання та аналіз великих даних;

- у випадку, якщо буде потрібно опрацювати дуже термінові операції, а FSB не зможе знайти відповідні ресурси для цих операцій, він надсилає терміновий запит до CSB процесу.

Однак, перед надсиланням цього запиту він інформує CSB про цей терміновий запит, позначає запит як завдання високого пріоритету. CSB, у свою чергу, застосовує механізм WFQ для класифікації, яка допомагає прискорити обробку пріоритетних запитів FSB.

Ще один момент, який слід зазначити, це те, що в деяких випадках Fog Gateway може мати можливість спілкуватися безпосередньо з CSB без необхідності спілкуватися з спочатку FSB. Така ситуація може мати місце у випадку роботи, яка менш чутлива до часу затримки.

2.2.4 Постачальник хмарних послуг

Постачальник хмарних послуг включає віртуальні та фізичні машини (ЦП, пам'ять, оперативна пам'ять, пропускна здатність, тощо). Кожна фізична машина може мати більше однієї віртуальної машини, і кожна віртуальна машина має агента який відповідає за атрибути роботи, отримані від хмарного брокера та інформує Job Monitor за допомогою повідомлення про стан завдання. Іншими словами, агент відповідає за моніторинг виконання завдань у хмарних провайдерів.

2.3 Висновки по другому розділу

У другому розділі було проведено аналіз сучасних методів управління перевантаженнями в розподілених інформаційних системах, зокрема в хмарних та туманних обчисленнях. Визначено, що існуючі підходи мають ряд обмежень, серед яких висока затримка обробки запитів, обмежена масштабованість та неефективне використання обчислювальних ресурсів. Проведений аналіз літератури показав, що найбільш перспективними підходами є використання механізмів динамічного балансування навантаження, адаптивного керування ресурсами та застосування алгоритмів штучного інтелекту для прогнозування навантажень. Встановлено, що розробка нових методів управління перевантаженнями дозволить підвищити продуктивність IoT-інфраструктури та покращити якість обслуговування.

Було запропоновано систему динамічного управління перевантаженнями (DCMB), що поєднує механізми пріоритетного обслуговування запитів та адаптивного балансування навантаження.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Опис методу організації експериментів

Для оцінки запропонованої нами системи були використані iFogSim і CloudSim. iFogSim є однією з основних платформ симуляції для обчислювальних середовищ туманних обчислень. CloudSim є одним з основних інструментів для моделювання хмарного середовища [49].

У таблицях 3.1 і 3.2 показано параметри для залучених фізичних і віртуальних машин в симуляції. Симуляція була розділена на дві частини з однаковими параметрами.

Таблиця 3.1 – Параметри віртуальних машин

Параметр	Значення параметру
Quantity	3
VM types	4
CPU processing power	1000
RAM space	512
Frequency range (Mbit/s)	100
Storage size (Gb)	5

Перша симуляція була розділена на дві частини. На обох 500 завдань від 5 користувачів з туману і хмарної системи (по 100 кожна) були відправлені в CSB. Моделювання передбачало наявність чотирьох машин (три віртуальні машини, побудовані на одній фізичній машині). Вони використовувались як постачальники. До 300 хмарних запитів можуть бути прийняті CSB, але запити, що перевищують цей ліміт, відхиляються. Період симуляції становив 24 години на основі інтервалів планування 300 с. У

таблиці 3.3 наведені характеристики хмарних програм (завдання, які обробляють запити), надісланих до CSB.

Таблиця 3.2 – Параметри фізичних машин

Параметр	Значення параметру
VM types	4
CPU processing power	1000
RAM space	2048
Frequency range (Mbit/s)	1000
Storage size (Gb)	10

Таблиця 3.3 – Параметри хмарних програм

Параметр	Значення параметру
Number	400
Job length	1000
File volume	300
Output volume	300
Number of PEs	2

Користувачі хмари або туману не надавали жодних платних послуг для високого або середнього пріоритету у першому розділі симуляції. Кожен користувач хмари представлений локальним агентом (4 користувача), а FSB (1 користувач) призначає «низький» пріоритет для кожного завдання. Характеристики FSB та пристрої IoT (сенсори), що використовуються в моделюванні, показані в таблицях 4–6.

У другому експерименті роботам FSB було присвоєно «високий» пріоритет, а без пріоритету було призначено користувачам хмари. Кожен тест вимірював загальний час обробки та відправки всіх завдань одночасно в CSB. Через велику кількість запитів, також було перевірено кількість затриманих або відхилених робіт.

Таблиця 3.4 – Характеристики Fog Service Broker (FSB)

Параметр	Значення
Вихідна пропускна здатність	1000
Вхідна пропускна здатність	1200
Процесор (MIPS)	5000
Ємність RAM	45000
Продуктивність (MIPS)	1000

Таблиця 3.5 – Характеристики першого сенсору

Параметр	Розмір/якість
Тип	EEG
Закон розподілення	Normal
Середнє значення	10
Стандартне відхилення	5

Таблиця 3.6 – Характеристики другого сенсору

Параметр	Розмір/якість
Закон розподілення	Uniform
Мінімальне значення	Normal
Максимальне значення	10

Було порівняно та оцінено обидва розділи експерименту за кількома факторами, в тому числі скільки часу знадобилося для виконання кожного завдання та скільки завдань було відхилено.

3.2 Опис моделювання

Було проведено декілька експериментів з моделювання різних ситуацій. Перший експеримент складався з двох симуляцій.

Перша симуляція: чотири користувача хмари надіслали 500 завдань до CSB, а один користувач хмари (користувач 5) надіслав 100 робіт до ФСБ. На цьому етапі не було призначено пріоритети, а запити FIFO були отримано та оброблено центральним процесором. Незалежно від порядку, в якому запитував користувач, перші 300 хмарних запитів від перших трьох користувачів були оброблені без будь-яких відмов. Оскільки CSB досяг межі прийняття на роботу (відповідно до своєї політики), роботи четвертого та п'ятого користувачів були відхилені. Обидва користувачі повторно надіслали свої завдання пізніше, щоб зберегти роботу у звичайному пріоритеті. На рисунку 3.1 показано час обробки, а на малюнку 3.2 показано відхилені роботи.

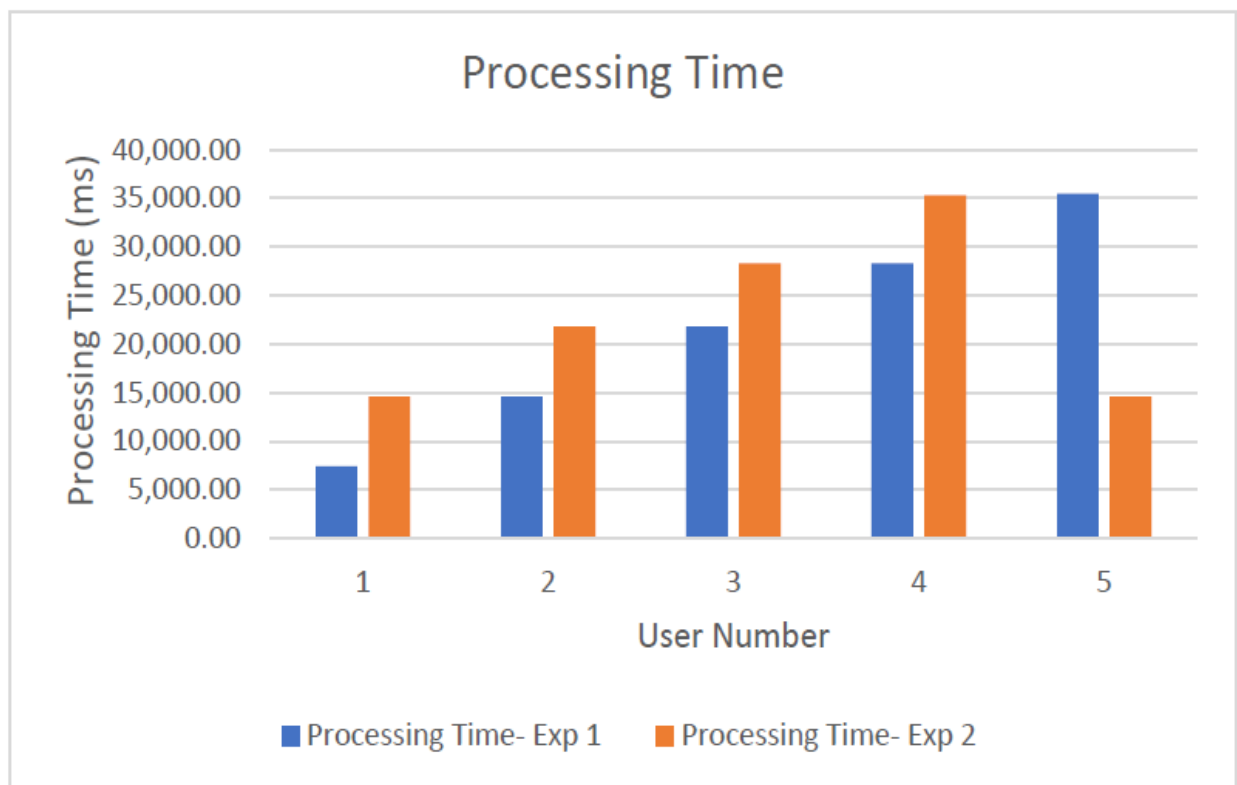


Рисунок 3.1 – Час, необхідний для обробки обох симуляцій першого експерименту

Друга симуляція: ще 500 запитів було надіслано до CSB з того самого джерела, але від ФСБ цього разу позначив свої завдання для користувачів як

пріоритетні. Інші користувачі хмари зберегли нормальний пріоритет. Як і у першому експерименті, рисунок 3.1 відображає час, що потребувала обробка запиту, а рисунку 3.2 містить значення відхилених завдань для першої та другої частин симуляції.



Рисунок 3.2 – Відмови запитів для обох симуляцій першого експерименту

Усі завдання були оброблені для користувача 5 ФСБ, що свідчить про переваги використання пріоритетних послуг. Крім того, у порівнянні з першою симуляцією, час обробки був дуже коротким. Більше того, відмови у роботі ФСБ не було. Використання системи DCMB давала гарантію виконання всіх робіт в короткі терміни. На основі механізму FIFO CSB, у користувачів 1 і 2 все ще не було відхилених завдань, тоді як у користувачів 3 і 4 було відхилено всі завдання. Таким чином, можна наглядно перевірити наскільки система справляється з потоками запитів.

Був проведений інший експеримент із тими ж налаштуваннями для фізичної машини, віртуальні машини, характеристики запитів (завдань) і характеристик пристроїв FSB та IoT (сенсорів), що було використано у першому експерименті моделювання (таблиці 1–6). П'ятсот робочих місць по 100 від чотирьох хмарних користувачів і 100 від ФСБ. Крім того, CSB прийняв до 300 запитів, і будь-які запити, що перевищували цю кількість, відхилялися. Відхилені запити можна було надіслати повторно через деякий інтервал часу.

Інтервал 300с. був використаний щодо дискретизації модельного часу для розподіленої системи у реальному часі моделювання 24 години. І знову симуляція була розділена на дві частини, які оцінювалися в з точки зору кількох факторів, таких як час виконання та відсоток відмов у обслуговуванні з боку хмари або туману. Час проведення моделювання складав 345 секунд.

3.3 Опис результатів моделювання

Перша симуляція: CSB отримав 400 завдань від хмарних користувачів 1–4 і 100 від FSB (користувач 5), без призначення пріоритету. CSB отримував роботи та керував ними за дисципліною обслуговування FIFO. Перші 300 завдань, які надійшли від перших трьох користувачів, були оброблені без жодних відмов від роботи. Згідно з його політикою, коли робочі місця четвертого та п'ятого користувачів досягли межі, що обмежена рівнем кількості вузлів, вони були відхилені CSB. Обробка зайняла більше часу у двох користувачів, їхні запити було оброблено після того, як вони їх повторно надіслали. За повторну передачу запита відповідає локальний брокер ресурсів, який наближений до користувачів або приборів IoT. На рисунках 3.3 і 3.4 відповідно показано обробку часу на вузлах системи та відхилення для експерименту, аналогічних тим, що було раніше описано у першому експерименті.

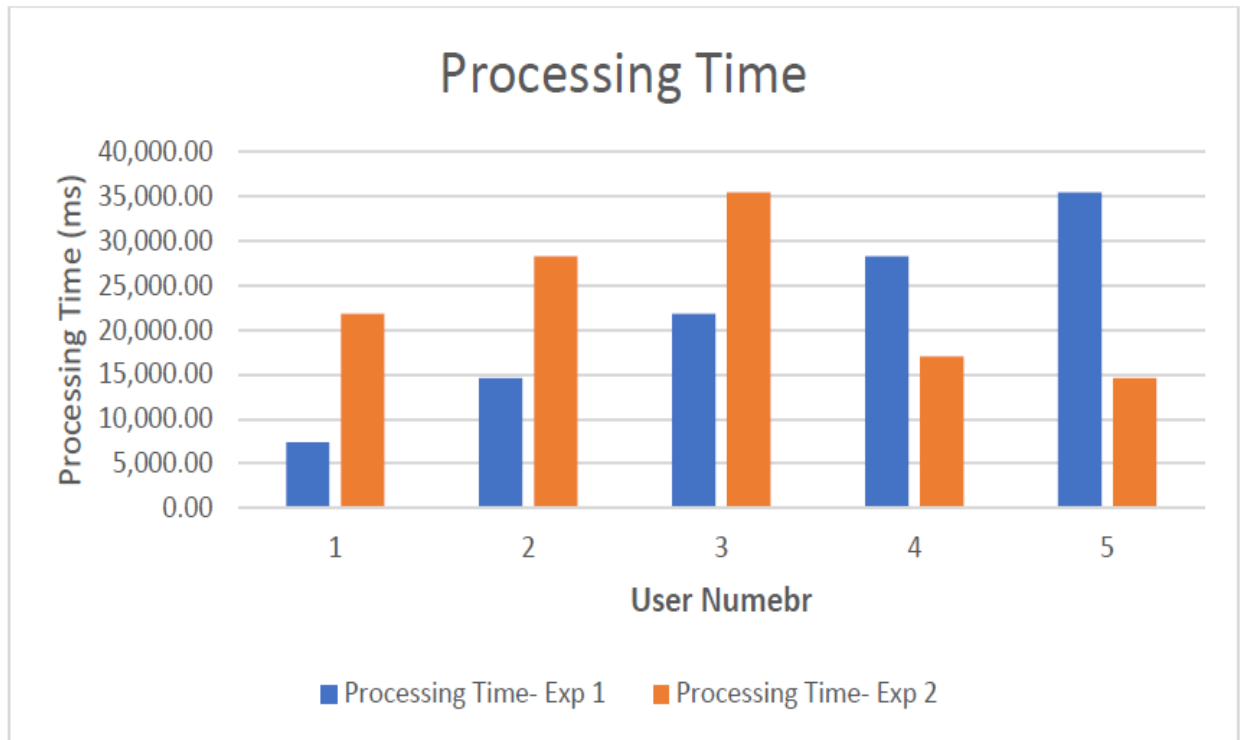


Рисунок 3.3 – Час, необхідний для обробки обох симуляцій другого експерименту

Друга симуляція: CSB отримував 400 завдань від хмарних користувачів 1–4 і 100 завдань від FSB (користувач 5). Для четвертого користувача місцевий агент позначив завдання як "середній" пріоритет (як платна послуга) а для користувача 5 ФСБ відзначила їх як «високий» пріоритет. Інші користувачі зберегли звичайний (тобто «низький») пріоритет. На рисунку 3.3 показано час обробки, а на малюнку 3.4 – відхилення завдань для моделювання.

Результати на малюнках 3.3 і 3.4 показують, що користувачі 5 (FSB) і 4 (хмарний користувач), які користувалися платною пріоритетною послугою, запровадженою системою DCMB, мали всі свої запити обробленими, і це було досягнуто за менший час порівняно з першим етапом експерименту. Для цих двох користувачів не було відкладених завдань та запитів. Це відбувалося тому, що CSB обробили свої завдання згідно дисципліни обслуговування FIFO. Перший користувач не мав відхилених запитів.

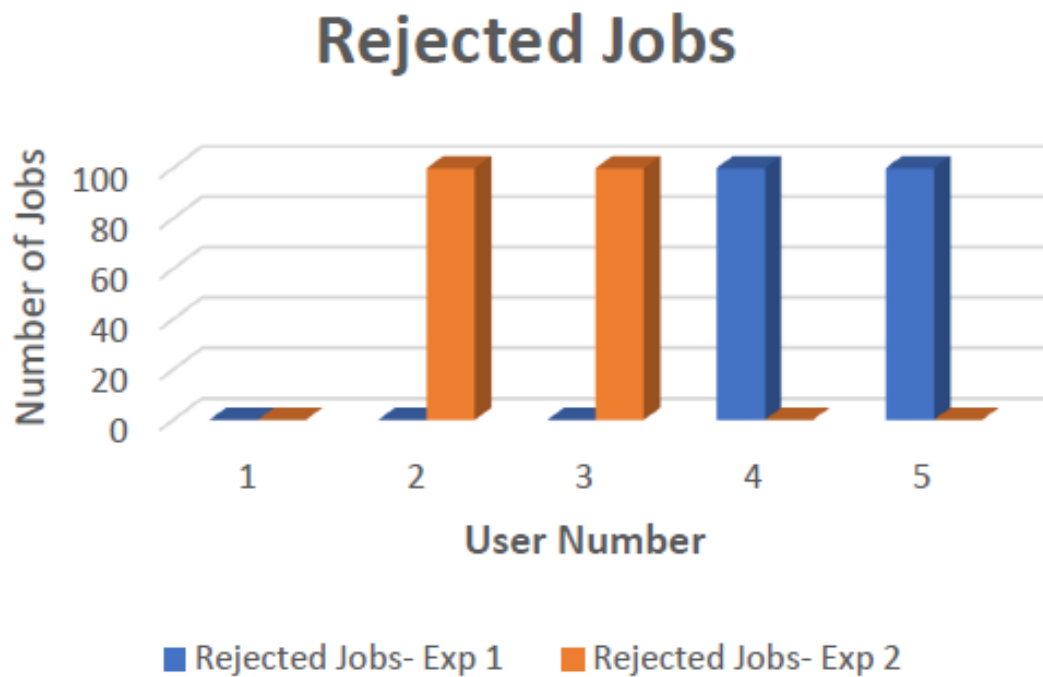


Рисунок 3.4 – Відмови запитів для обох симуляцій другого експерименту

3.4 Висновки по третьому розділу

Проведено тестування ефективності запропонованої системи в середовищах iFogSim та CloudSim, що дозволило оцінити вплив розроблених алгоритмів на продуктивність IoT-систем. Результати експериментів показали, що використання DCMB дозволяє зменшити середній час обробки запитів, знизити рівень перевантаження центральних вузлів та покращити ефективність використання ресурсів. Отримані результати підтверджують доцільність впровадження DCMB у реальні IoT-мережі для забезпечення стабільної та ефективної роботи систем у умовах високих навантажень.

ВИСНОВКИ

Програми IoT швидко трансформують традиційні моделі хмарних обчислень, які зі швидким розвитком стикаються з такими проблемами, як пропускна здатність, затримка та збої мережі додатків IoT та зростання вимог і очікувань від користувачів. Тумані обчислення зазвичай обробляють і зберігають дані локально між пристроями IoT замість надсилання їх до хмари. Однак у випадках, коли є необхідність надіслати деякі конфіденційні дані для їх обробки потрібен швидкий час відповіді на хмару, величезну кількість хмарні запитів, що надходять із рівня посередника туману, можуть перевантажувати приховані ресурси і мати технічні проблеми та знижений QoS.

У кваліфікаційній роботі представлено систему (DCMB), яка задовольняє вимогам IoT SLA для QoS [50]. Результати проведених експериментів показують, що користувачі із середовища туману та інші користувачі хмарного середовища, які використовували його на платній пріоритетній основі, що запроваджена системою DCMB, дозволила обробити всі їхні завдання, і це було досягнуто за менший час (15 000 мс) порівняно з першим етапом експерименту (35 000 мс). У рамках виконання завдання також надавалася і відстежувалася робота хмарних провайдерів.

Запропонована система DCMB була оцінена за допомогою iFogSim і CloudSim, і результати показали, що використання DCMB підвищило ступінь, до якого вимоги QoS IoT були дотримані під час використання ресурсів і хмари уникали порушення SLA [49].

Експерименти показують, що час обробки, необхідний системі DCMB, був дуже коротким порівняно з традиційною хмарною системою та кількістю відхилених запитів, отриманих у результаті використання системи DCMB, нижча, ніж у традиційній хмарній системі.

Запропонована система має певні обмеження, як-от її відсутність у великій кількості робочих місць. В експериментах було задіяно лише 500 робочих місць. Щоб перевірити його придатність і продуктивність, планується розширити запропоновану систему та протестувати її за допомогою великої кількості запитів.

Практична значущість роботи полягає у наступному

- запропонована модель дозволяє ефективніше розподіляти ресурси в гетерогенних IoT-системах;
- використання трирівневої архітектури спрощує масштабування та інтеграцію нових пристроїв;
- поліпшена синхронізація станів об'єктів знижує затримки в обміні даними;
- результати можуть бути застосовані для моделювання систем розумних міст, автоматизованого управління виробництвом, транспортних мереж тощо.

Перспективні напрямки подальших досліджень включають розробку адаптивних механізмів балансування навантаження у масштабних розподілених системах, інтеграцію методів штучного інтелекту для прогнозування поточного навантаження мереж Інтернет речей, а також вдосконалення алгоритмів синхронізації для стабільної роботи в умовах високого навантаження.

Одним із ключових напрямків є використання технологій штучного інтелекту для прогнозування великих навантажень та автоматичної оптимізації характеристик інформаційних систем.

Окрему увагу слід приділити підвищенню рівня безпеки IoT-інфраструктури. Для цього можна використовувати сучасні методи шифрування та аутентифікації.

Додатково, технології блокчейну можуть забезпечити підвищений рівень надійності та безпеки обміну даними між пристроями, тоді як

впровадження самонавчальних механізмів сприятиме підвищенню рівня незалежності систем Інтернету речей.

Запропоновані дослідження сприятимуть збільшенню функціональних можливостей розподілених IoT-систем та формуванню заходів для їх сталого постійного розвитку. У перспективі особливо важливою є автоматизація процесів управління ресурсами, що дозволить мінімізувати залежність системи від втручання людини та підвищити її ефективність.

Таким чином, можна зробити висновок, що усі завдання та мета роботи виконані у повному обсязі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Alouffi, B.; Hasnain, M.; Alharbi, A.; Alosaimi, W.; Alyami, H.; Ayaz, M. A Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies. *IEEE Access* 2021, 9, 57792–57807.
2. Волк М.О., Єрохін Б.О. Вибір технологічного стеку розробки приватної хмарної системи паралельного збору та обробки інформації. *Вісник Херсонського національного технічного університету*. No 1(84), 2023 с. 134-139. DOI: <https://doi.org/10.35546/kntu2078-4481.2023.1.18>.
3. Stergiou, C.; Psannis, K.E.; Gupta, B.B.; Ishibashi, Y. Security, privacy & efficiency of sustainable Cloud Computing for Big Data & IoT. *Sustain. Comput. Inform. Syst.* 2018, 19, 174–184.
4. Mbarek, N. Service Level Management in the Cloud. *Serv. Level Manag. Emerg. Environ.* 2021, 20, 45–81.
5. Washizaki, H.; Xia, T.; Kamata, N.; Fukazawa, Y.; Kanuka, H.; Kato, T.; Yoshino, M.; Okubo, T.; Ogata, S.; Kaiya, H.; et al. Systematic Literature Review of Security Pattern Research. *Information* 2021, 12, 36.
6. Nayyar, A. *Handbook of Cloud Computing: Basic to Advance Research on the Concepts and Design of Cloud Computing*; BPB Publications: Noida, India, 2019.
7. Butpheng, C.; Yeh, K.H.; Xiong, H. Security and privacy in IoT-cloud-based e-health systems—A comprehensive re-view. *Symmetry* 2020, 12, 1191.
8. De Sousa, N.F.; Perez, D.A.; Rosa, R.V.; Santos, M.A.; Rothenberg, C.E. Network service orchestration: A survey. *Comput. Commun.* 2019, 142, 69–94.
9. Alwada'n, T. Mobility in Cloud Systems. *Int. J. Comput. Trends Technol. (IJCTT)* 2015, 11, 202–205.
10. Bello, S.A.; Oyedele, L.O.; Akinade, O.O.; Bilal, M.; Delgado, J.M.D.; Akanbi, L.A.; Ajayi, A.O.; Owolabi, H.A. Cloud computing in construction industry: Use cases, benefits and challenges. *Autom. Constr.* 2021, 122, 103441.

11. Services, A.W. Aws Economics Center. May 2012. Available online: <https://aws.amazon.com/economics/> (accessed on 28 October 2022).
12. Sanger, A.K.; Johari, R. Survey of Security Issues in Cloud. In Proceedings of the 2022 International Mobile and Embedded Technology Conference (MECON), Noida, India, 10 March 2022; pp. 490–493.
13. The Office of the Privacy Commissioner of Canada, O.: Introduction to Cloud Computing. October 2011. Available online: https://www.priv.gc.ca/resource/fs-fi/02_05_d_51_cc_e.asp (accessed on 28 October 2022).
14. Chawla, I. Cloud Computing Environment: A Review. *J. Int. J. Comput. Technol.* 2018, 17.
15. Jamsa, K. Cloud Computing; Jones & Bartlett Learning Publisher: Burlington, VT, USA, 2022; ISBN 9781284233971.
16. Alwada'n, T.; Al-Zitawi, O.; Atoum, J. Cloud Computing: Privacy, Mobility and Resources Utilization. *Int. J. Comput. Trends Technol. (IJCTT)* 2016, 41, 29–36.
17. Mohammad, H.; Zaman, K.R. A Systematic Review on Cloud Computing. *Int. J. Comput. Sci. Eng.* 2018, 6, 632–639.
18. Alwada'n, T. Cloud Computing Topology: Towards Enhancing the Performance. *Int. J. Comput. Sci. Inf. Secur.* 2016, 14, 654–658.
19. Alwada'n, T. Cloud Computing and Multi-Agent System: Monitoring and Services. *J. Theor. Appl. Inf. Technol.* 2018, 96, 2435–2444.
20. Ahluwalia, J.K.; Mouradian, C.; Alam, M.N.; Glitho, R. A Cloud Infrastructure as a Service for an Efficient Usage of Sensing and Actuation Capabilities in Internet of Things. In Proceedings of the NOMS 2022–2022 IEEE/IFIP Network Operations, Budapest, Hungary, 25–29 April 2022; pp. 1–6.
21. Quattra Inc. Quattra Cloud Vision and Frame Work Value. March 2017, pp. 1–5. Available online: <https://silo.tips/download/quattra-s-cloud-vision-framework-value> (accessed on 8 December 2022).
22. Yasrab, R. Platform-as-a-Service (PaaS): The Next Hype of Cloud

Computing. arXiv 2018, arXiv:1804.10811.

23. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutorials* 2015, 17, 2347–2376.

24. Nalin, M.; Baroni, I.; Mazzara, M. A Holistic Infrastructure to Support Elderlies' Independent Living. In *Encyclopedia of E-Health and Telemedicine*; IGI Global: Hershey, PA, USA, 2016.

25. Salikhov, D.; Khanda, K.; Gusmanov, K.; Mazzara, M.; Mavridis, N. Microservice-based IoT for Smart Buildings. In *Proceedings of the 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Taipei, Taiwan, 27–29 March 2017.

26. Волк М.О., Бугрій А.М., Самойлов І.А., Фурманов А.А., Журавльов О.Ю., Волк Д.М. Симуляція та управління туманними та хмарними обчисленнями для інтернету речей. *Вісник Херсонського національного технічного університету*. No 3(90), 2024 с. 215-220. DOI: <https://doi.org/10.35546/kntu2078-4481.2024.3.27>

27. De Donno, M.; Giarretta, A.; Dragoni, N.; Bucchiarone, A.; Mazzara, M. Cyber-Storms Come from Clouds: Security of Cloud Computing in the IoT Era. *Futur. Internet* 2019, 11, 127.

28. Yashodhan, A.; Sridhar, K. A Device-Independent Efficient Actigraphy Signal-Encoding System for Applications in Monitoring Daily Human Activities and Health. *Sensors* 2018, 18, 2966.

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

30. Ai, Y.; Peng, M.; Zhang, K. Edge cloud computing technologies for Internet of things: A primer. *Digit. Commun. Netw.* 2017; in press.

31. Atlam, H.F.;Walters, R.J.;Wills, G.B. Fog Computing and the Internet of Things: A Review. *Big Data Cogn. Comput.* 2018, 2, 10.
- Wang, X.; Han, Y.; Leung, V.C.M.; Niyato, D.; Yan, X.; Chen, X. Convergence of Edge Computing and Deep Learning: A Comprehensive Survey. *IEEE Commun. Surv. Tutorials* 2020, 22, 869–904.
33. Peter, N. FOG Computing and Its Real Time Applications. *Int. J. Emerg. Technol. Adv. Eng.* 2015, 5, 266–269.
34. Wen, Z.; Yang, R.; Garraghan, P.; Lin, T.; Xu, J.; Rovatsos, M. Fog Orchestration for Internet of Things Services. *IEEE Internet Comput.* 2017, 21, 16–24.
35. Yang, Y. FA2ST: Fog as a Service Technology. In *Proceedings of the 2017 IEEE 41st IEEE Annual Computer Software and Applications Conference*, Turin, Italy, 4–8 July 2017; p. 708.
36. Chiang, M.; Zhang, T. Fog and IoT: An Overview of Research Opportunities. *IEEE Internet Things J.* 2016, 3, 854–864.
37. Cisco, Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. April 2015. Available online: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf (accessed on 14 July 2020).
38. Al-Ayyoub, M.; Jararweh, Y.; Daraghmeh, M.; Althebyan, Q. Multi-agent based dynamic resource provisioning and monitoring for cloud computing systems infrastructure. *Clust. Comput.* 2015, 18, 919–932.
39. Goudarzi,M.; Ilager, S.; Buyya, R. Cloud Computing and Internet of Things: Recent Trends and Directions. In *New Frontiers in Cloud Computing and Internet of Things*; Springer International Publishing: Cham, Switzerland, 2022; pp. 3–29. ISBN 978-3-031-05528-7.
40. Dastjerdi, A.V.; Gupta, H.; Calheiros, R.N.; Ghosh, S.K.; Buyya, R. Fog Computing: Principles, architectures, and applications. In *Internet of Things: Principles and Paradigms*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2016; pp. 61–75.

41. Suárez-Albela, M.; Fernández-Caramés, T.M.; Fraga-Lamas, P.; Castedo, L. A practical evaluation of a high-security energy-efficient gateway for IoT fog computing applications. *Sensors* 2017, 17, 1978. [PubMed]
42. Kong, L.; Tan, J.; Huang, J.; Chen, G.; Wang, S.; Jin, X.; Zeng, P.; Khan, M.K.; Das, S.K. Edge-Computing-Driven Internet of Things: A Survey. *ACM Comput. Surv.* 2022; Just Accepted.
43. Alwada'n, T.; Al-Tamimi, A.; Mohammad, A.H.; Salem, M.; Muhammad, Y. Dynamic Congestion Management System for Cloud Service Broker. *Int. J. Electr. Comput. Eng. (IJECE)*. Febr. 2023, 13, 872–883.
44. Bellavista, P.; Berrocal, J.; Corradi, A.; Das, S.K.; Foschini, L.; Zanni, A. A survey on fog computing for the internet of things. *Pervasive Mob. Comput.* 2019, 52, 71–99.
45. Pérez, J.; Díaz, J.; Berrocal, J.; López-Viana, R.; González-Prieto, Á. Edge computing. *Computing* 2022, 104, 2711–2747.
46. Yousefpour, A.; Fung, C.; Nguyen, T.; Kadiyala, K.; Jalali, F.; Niakanlahiji, A.; Kong, J.; Jue, J.P. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *J. Syst. Archit.* 2019, 98, 289–330.
47. Kubiak, K.; Dec, G.; Stadnicka, D. Possible Applications of Edge Computing in the Manufacturing Industry—Systematic Literature Review. *Sensors* 2022, 22, 2445.
48. Волк М.О., Поповкін М. М. Методи моделювання масштабованих хмарних ресурсів. Системи управління, навігації та зв'язку, 2024, випуск 3(77), с. 56-59. doi:<https://doi.org/10.26906/SUNZ.2024.3.097>
49. Волк М.О., Бугрій А.М., Самойлов І.А., Фурманов А.А., Журавльов О.Ю., Волк Д.М. Симуляція та управління туманними та хмарними обчисленнями для інтернету речей. Вісник Херсонського національного технічного університету. No 3(90), 2024 с. 215-220. DOI: <https://doi.org/10.35546/kntu2078-4481.2024.3.27>