

МАГІСТЕРСЬКА РОБОТА

МР. ШМ - 18.00.00.000 ПЗ

Група ШМ-24-1

Дрогомирецький Назарій

2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Дрогомирецький Назарій Васильович

(прізвище, ім'я, по батькові)

УДК 004.9
(індекс)

МАГІСТЕРСЬКА РОБОТА

Моделі та методи реалізації транспортних мереж на основі

концепції великих даних

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Дрогомирецький Н.В.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник **Юрчишин Володимир Миколайович, д.т.н., професор**

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту

Завідувач кафедри

доц. **Бандура В.В.**

(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль

доц. **Вовк Р.Б.**

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою

ІПЗ

доц.

В.В. Бандура

“ 04 ” вересня 2025 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Дрогомирецькому Назарію Васильовичу

(прізвище, ім'я, по-батькові)

1. Тема магістерської роботи “Моделі та методи реалізації транспортних мереж на основі концепції великих даних”

керівник проекту (роботи) Юрчишин В.М., д.т.н., професор

затверджені наказом закладу вищої освіти від “ 05 ” листопада 2025 р. № 695/7

2. Строк подання студентом проекту (роботи) 15 грудня 2025 р.

3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови інформаційних технологій обробки великих даних

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

1. Аналіз предметної області застосування транспортних мереж для обробки даних

2. Методи та концепції реалізації транспортних мереж на основі концепції великих даних

3. Концептуальна архітектура та моделювання автомобільного дата-центру

4. Застосування моделей та методів реалізації транспортних мереж на основі великих даних

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Архітектура Vehicular Ad-hoc Network (рис. 1.1)

2. Потік даних Edge Computing (рис. 1.2)

3. Ключові компоненти та їх взаємодія в Smart Vehicle (рис. 1.3)

4. Графічна інтерпретація алгоритму MapReduce (рис. 1.4)

5. Моделі хмарного обслуговування (рис. 2.1)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2025 р.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерського дослідження	15.09.2025	виконано
2	Аналіз предметної області застосування транспортних мереж для обробки даних	29.09.2025	виконано
3	Методи та концепції реалізації транспортних мереж на основі концепції великих даних	15.10.2025	виконано
4	Концептуальна архітектура та моделювання автомобільного дата-центру	08.11.2025	виконано
5	Застосування моделей та методів реалізації транспортних мереж на основі великих даних	16.11.2025	виконано
6	Затвердження пояснювальної записки роботи завідувачем кафедри	14.12.2025	виконано

Студент – магістр _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Магістерська робота: 76 с., 20 рис., 4 табл., 40 джерел.

Тема: Моделі та методи реалізації транспортних мереж на основі концепції великих даних

Мета роботи - розроблення моделей і методів реалізації транспортних мереж на основі концепції великих даних, що забезпечують ефективну обробку, розподіл і використання обчислювальних ресурсів у динамічних мобільних середовищах.

Об'єкт дослідження - процеси обробки та управління великими даними в транспортних мережах, побудованих на основі технологій розподілених обчислень.

Предмет дослідження - моделі, методи та архітектурні рішення реалізації транспортних мереж, орієнтованих на інтеграцію концепції великих даних у розподілені обчислювальні середовища.

Результати дослідження

В роботі виконано побудову концептуальної архітектури дата-центру як розподіленої обчислювальної системи для обробки великих даних, що інтегрує підходи хмарних технологій, віртуалізації та мережевої взаємодії транспортних засобів.

Висновок

Досліджено та створено модель симуляційного середовища для аналізу продуктивності транспортних обчислювальних мереж, що може бути використана в подальших наукових і прикладних дослідженнях.

**ТРАНСПОРТНА МЕРЕЖА; ВЕЛИКІ ДАНІ; АТА-ЦЕНТР;
ІНТЕРНЕТ РЕЧЕЙ; РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ; ВІРТУАЛІЗАЦІЯ;
СИМУЛЯЦІЙНЕ МОДЕЛЮВАННЯ; ІНТЕЛЕКТУАЛЬНІ ТРАНСПОРТНІ СИСТЕМИ.**

ABSTRACT

Master Thesis: 76 pp., 20 fig., 4 tab., 40 sources.

Topic: Models and methods for implementing transport networks based on the concept of big data

The purpose of the work is to develop models and methods for implementing transport networks based on the concept of big data, which ensure effective processing, distribution and use of computing resources in dynamic mobile environments.

The object of the study is the processes of processing and managing big data in transport networks built on the basis of distributed computing technologies.

The subject of the study is models, methods and architectural solutions for implementing transport networks focused on integrating the concept of big data into distributed computing environments.

Research results

The work builds a conceptual architecture of a data center as a distributed computing system for processing big data, which integrates approaches of cloud technologies, virtualization and network interaction of vehicles.

Conclusion

A model of a simulation environment for analyzing the performance of transport computing networks has been studied and created, which can be used in further scientific and applied research.

TRANSPORT NETWORK; BIG DATA; ATA CENTER; INTERNET OF THINGS; DISTRIBUTED COMPUTING; VIRTUALIZATION; SIMULATION MODELING; INTELLIGENT TRANSPORT SYSTEMS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	10
ВСТУП.....	11
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ ТРАНСПОРТНИХ МЕРЕЖ ДЛЯ ОБРОБКИ ДАНИХ	15
1.1. Використання транспортних мереж як розподілених обчислювальних ресурсів.....	15
1.1.1. Методологія розробки дата-центру	16
1.1.2. Архітектура VANET мережі	17
1.2. Концептуалізація розподілених обчислювальних систем на базі інтернету речей	18
1.2.1. Еволюція та масштаби інтернету речей (IoT)	18
1.2.2. Феномен крайових обчислень (Edge Computing).....	19
1.2.3 Інтелектуальний транспортний засіб як обчислювальний вузол	21
1.2.4. Методологія моделювання	22
1.3. Інтернет речей (IoT) в контексті розвитку транспортних мереж	23
1.3.1. Роль розумних автомобілів у транспортних мережах	23
1.3.2. Критичні виклики безпеки в екосистемі IoT	26
1.4. Науковий аналіз зростання великих даних та розподілені обчислювальні стратегії	27
1.4.1. Вимоги до обробки великих даних	27
1.4.2. Парадигма розподілених обчислень (Distributed Computing). Принцип MapReduce	28
Висновки до розділу	31
РОЗДІЛ 2. МЕТОДИ ТА КОНЦЕПЦІЇ РЕАЛІЗАЦІЇ ТРАНСПОРТНИХ МЕРЕЖ НА ОСНОВІ КОНЦЕПЦІЇ ВЕЛИКИХ ДАНИХ.....	32

2.1. Консолідація ресурсів та моделі обслуговування в парадигмі хмарних обчислень	32
2.1.1. Моделі хмарного обслуговування.....	32
2.1.2. Дата-центр як основа хмарних обчислень.....	34
2.2. Огляд досліджень Vehicular Cloud Computing (VCC) від концепції до архітектури дата-центру	35
2.2.1. Концепція автомобільних хмар (Vehicular Clouds).....	35
2.2.2. Моделювання доступності ресурсів	37
2.2.3. IoT-орієнтовані сервіси та обробка великих даних	38
2.3. Концептуальна архітектура та моделювання автомобільного дата-центру	42
2.3.1. Концепція автомобільного дата-центру (VDC).....	42
2.3.2. Архітектура обчислювальних ресурсів (віртуалізація)	43
2.3.3. Характеристики завдань обробки великих даних.....	44
Висновки до розділу	45

РОЗДІЛ 3. ЗАСТОСУВАННЯ МОДЕЛЕЙ ТА МЕТОДІВ РЕАЛІЗАЦІЇ

ТРАНСПОРТНИХ МЕРЕЖ НА ОСНОВІ ОБРОБКИ ВЕЛИКИХ ДАНИХ ...	47
3.1. Архітектура симуляції хмарного середовища (VDC) обробки великих даних.....	47
3.1.1 Контролер дата-центру (DC).....	47
3.2. Архітектура та моделювання мережі дата-центру (VDC) транспортної мережі.....	49
3.2.1. Моделювання вузлів	49
3.2.2. Ієрархічна топологія мережевого дерева.....	50
3.2.3. Моделювання мережевого трафіку.....	51
3.2.4. Класифікація мережевих комунікацій.....	52
3.3. Архітектура віртуалізації обчислювальних вузлів автомобільної хмари	53

3.4. Методологія симуляції життєздатності дата-центру транспортної мережі як механізму обробки великих даних	54
3.4.1 Фактори та константи симуляції.....	54
3.4.2 Деталізація змінних	55
3.5. Експериментальна оцінка проведеного стимуляційного експерименту.....	57
3.5.1. Оптимізація кількості робітників (Workers).....	57
3.5.2. Оцінка стаціонарного стану часу виконання завдань.....	59
3.5.3. Порівняння продуктивності: випадкові vs. фіксовані розміри завдань.....	62
3.5.4. Вплив мережевої топології.....	64
3.5.5. Порівняння з традиційним дата-центром	66
Висновки до розділу	68
ВИСНОВКИ	70
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	73

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- VC - Vehicular Cloud - автомобільна хмара
- VCC - Vehicular Cloud Computing - автомобільні хмарні обчислення
- VDC - Vehicular Data Center - автомобільний дата-центр
- CaaS - Computing as a Service - обчислення як послуга
- SaaS - Storage as a Service - зберігання як послуга
- GT - Gateway Terminal - шлюзовий термінал
- DC - Data Center Controller - контролер дата-центру
- VM - Virtual Machine - Віртуальна Машина
- VMM - Virtual Machine Monitor - менеджер віртуальних машин (гіпервізор)
- DOE - Design of Experiments - планування експерименту

ВСТУП

Актуальність теми.

Сучасний розвиток інформаційних технологій характеризується стрімким зростанням обсягів даних, що генеруються у різноманітних галузях людської діяльності. Особливої актуальності набуває проблема ефективної обробки, зберігання та аналізу великих даних (Big Data) у розподілених і динамічних середовищах. Однією з таких сфер є інтелектуальні транспортні системи (ITS), де потоки інформації надходять від транспортних засобів, дорожньої інфраструктури, сенсорних мереж та мобільних пристроїв у реальному часі. Для забезпечення високої швидкості реагування, масштабованості та надійності обчислювальних процесів необхідно поєднувати технології великих даних, хмарних обчислень, інтернету речей (IoT) і транспортних мереж нового покоління.

У цьому контексті перспективним напрямом є розроблення моделей і методів реалізації транспортних мереж, здатних виконувати функції розподілених обчислювальних середовищ. Особливе місце займають Vehicular Ad-hoc Networks (VANETs) — самоорганізовані транспортні мережі, у яких автомобілі виступають як вузли, що взаємодіють безпосередньо один з одним і з інфраструктурою. Поєднання концепцій VANET, Vehicular Cloud Computing (VCC) та Big Data дозволяє створювати дата-центри (Vehicular Data Centers, VDC), у яких транспортні засоби динамічно об'єднують свої ресурси для обробки великих масивів інформації.

Розроблення ефективних моделей і методів для реалізації таких систем потребує теоретичного обґрунтування архітектури, методології моделювання, алгоритмів розподілу ресурсів та оцінки продуктивності. Ця робота спрямована на створення наукових і практичних основ побудови транспортних мереж, орієнтованих на обробку великих даних, що відповідають вимогам сучасних інтелектуальних транспортних систем.

Актуальність теми зумовлена глобальними тенденціями розвитку інтелектуальних транспортних систем, інтернету речей та технологій великих даних. Кількість транспортних засобів, обладнаних сенсорами, бездротовими модулями та навігаційними системами, зростає експоненційно, що призводить до виникнення колосальних обсягів інформації, яка потребує миттєвої обробки. Традиційні централізовані системи обчислень втрачають ефективність у таких умовах через обмеження пропускної здатності, затримки передачі даних і відсутність гнучкості.

У зв'язку з цим особливо важливо розробити нові підходи до організації транспортних мереж, які здатні динамічно розподіляти обчислювальні ресурси між рухомими об'єктами. Концепції VANET і Vehicular Cloud Computing відкривають нові можливості для створення адаптивних і самоорганізованих систем, де автомобілі виконують роль обчислювальних вузлів. Це дозволяє формувати мобільні дата-центри, що інтегрують обробку великих даних безпосередньо в транспортну інфраструктуру. Отже, дослідження моделей і методів реалізації транспортних мереж на основі концепції великих даних є важливим як у науковому, так і в практичному аспекті. Воно сприяє розвитку нових інформаційних технологій, здатних забезпечити ефективну обробку даних у реальному часі для транспортної галузі, логістики, екологічного моніторингу та систем автономного керування.

Метою роботи є розроблення моделей і методів реалізації транспортних мереж на основі концепції великих даних, що забезпечують ефективну обробку, розподіл і використання обчислювальних ресурсів у динамічних мобільних середовищах.

Об'єктом дослідження є процеси обробки та управління великими даними в транспортних мережах, побудованих на основі технологій розподілених обчислень.

Предметом дослідження є моделі, методи та архітектурні рішення реалізації транспортних мереж, орієнтованих на інтеграцію концепції великих даних у розподілені обчислювальні середовища.

Для досягнення поставленої мети в роботі необхідно виконати такі завдання:

- Провести аналіз предметної області застосування транспортних мереж у контексті великих даних і розподілених обчислень.
- Дослідити архітектурні підходи до побудови Vehicular Ad-hoc Networks та інтернету речей (IoT).
- Розробити концептуальну модель дата-центру для обробки великих даних у транспортному середовищі.
- Запропонувати методи моделювання розподілу ресурсів та організації обчислювальних процесів у Vehicular Cloud Computing.
- Створити симуляційне середовище для дослідження продуктивності транспортних обчислювальних мереж.
- Провести експериментальну оцінку ефективності запропонованих моделей та визначити фактори, що впливають на час виконання завдань і стабільність системи.

Методи дослідження

У роботі використано комплекс методів наукового аналізу, моделювання та симуляції. Теоретичною основою стали методи системного аналізу, теорії розподілених обчислень, теорії черг та мережевих топологій. Для побудови моделей застосовано методи імітаційного моделювання, паралельних обчислень і математичного опису динамічних систем. Експериментальна частина дослідження базується на використанні симуляційного середовища для оцінки параметрів продуктивності обчислювальних вузлів та мережевої інфраструктури транспортного дата-центру.

Наукова новизна отриманих результатів

Удосконалено підхід до побудови транспортних мереж на основі концепції великих даних шляхом інтеграції моделей VANET, IoT та Vehicular Cloud Computing. Запропоновано концептуальну архітектуру дата-центру (VDC), який реалізує принципи віртуалізації, динамічного балансування навантаження та паралельної обробки даних.

Практичне застосування результатів

Розроблені моделі та методи можуть бути використані для створення систем інтелектуального управління транспортними потоками, моніторингу дорожньої інфраструктури, логістичних мереж і автономного транспорту. Запропоноване симуляційне середовище може бути основою для подальших досліджень з оптимізації розподілених обчислень у мобільних середовищах.

Структура магістерської роботи. Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 76 сторінок, і містить 20 рисунків, 4 таблиці, список використаних джерел із 40 найменувань.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ ТРАНСПОРТНИХ МЕРЕЖ ДЛЯ ОБРОБКИ ДАНИХ

1.1. Використання транспортних мереж як розподілених обчислювальних ресурсів

Сучасні транспортні мережі (Vehicular Ad-hoc Networks, VANETs) стали повсюдним елементом транспортної інфраструктури, що зумовило розробку численних додатків, орієнтованих на використання їхніх сенсорних та обчислювальних ресурсів у динамічному стані. Однак, значний потенціал ресурсів залишається невикористаним у періоди стаціонарного перебування транспортних засобів. Припарковані автомобілі з підключенням до ширшої обчислювальної інфраструктури можуть ефективно інтегрувати свої бездіяльні ресурси в цю систему. Агрегація достатньої кількості таких взаємопов'язаних автомобілів формує розподілене обчислювальне середовище, здатне підтримувати виконання хмарних додатків. Для електромобілів (EVs) це також відкриває можливість інтеграції з послугами зарядки.

Vehicular Ad-hoc Networks (VANETs) — це тип мобільної ad-hoc мережі (MANET), спеціально розроблений для транспортних засобів, які можуть самостійно обмінюватися даними між собою (V2V — Vehicle-to-Vehicle) та з інфраструктурою (V2I — Vehicle-to-Infrastructure) без потреби у фіксованій базовій станції.

Іншими словами, VANET — це динамічна бездротова мережа транспортних засобів, яка забезпечує взаємодію автомобілів для підвищення безпеки руху, ефективності транспорту та комфорту пасажирів.

Основні характеристики VANET:

- транспортні засоби швидко рухаються, тому топологія мережі постійно змінюється.

- з'єднання між вузлами (автомобілями) можуть встановлюватися й розриватися за лічені секунди.
- VANET-и мають мінімальні затримки для обміну критичною інформацією (наприклад, попередження про аварію).
- маршрути часто визначаються на основі GPS-координат, а не за IP-адресами.
- високі вимоги до безпеки та конфіденційності, оскільки обмін даними не повинен порушувати приватність водіїв.

1.1.1. Методологія розробки дата-центру

Ця робота представляє розробку спрощеної архітектури автомобільного дата-центру (Vehicular Data Center) на основі інтелектуальних транспортних засобів, що перебувають на парковці.

Ключові відмінності від попередніх досліджень:

- Попередні моделі часто покладалися на міграцію завдань (task migration), зумовлену передбачуваним часом перебування (dwell time) автомобілів.
- Наша модель виходить з припущення непередбачуваності часу перебування, що унеможливорює ефективне застосування міграції завдань.

Для компенсації відсутності міграції завдань застосовано паралельний обчислювальний підхід "розділяй і володарюй" (Divide and Conquer). В основі цього підходу лежить використання парадигми MapReduce для:

1. Поділу завдання на множину незалежних підзавдань (subtasks).
2. Паралельної обробки цих підзавдань за допомогою ресурсів припаркованих автомобілів.

Для мінімізації втрати проміжних даних у разі несподіваного виходу автомобіля з системи (від'їзду) між фазами Map та Reduce реалізовано механізм контрольної точки (checkpointing).

Ця розроблена модель слугуватиме засобом для валідації практичної реалізованості концепції автомобільного дата-центру та стане базисом для

порівняння в майбутніх дослідженнях, орієнтованих на оптимізацію використання стаціонарних ресурсів транспортних мереж.

1.1.2. Архітектура VANET мережі

На рисунку 1.1 подано архітектуру Vehicular Ad-hoc Network (VANET), яка є ключовим елементом інтелектуальних транспортних систем (ITS). Вона демонструє трирівневу систему, що забезпечує комунікацію "Транспортний Засіб до Усього" (V2X).

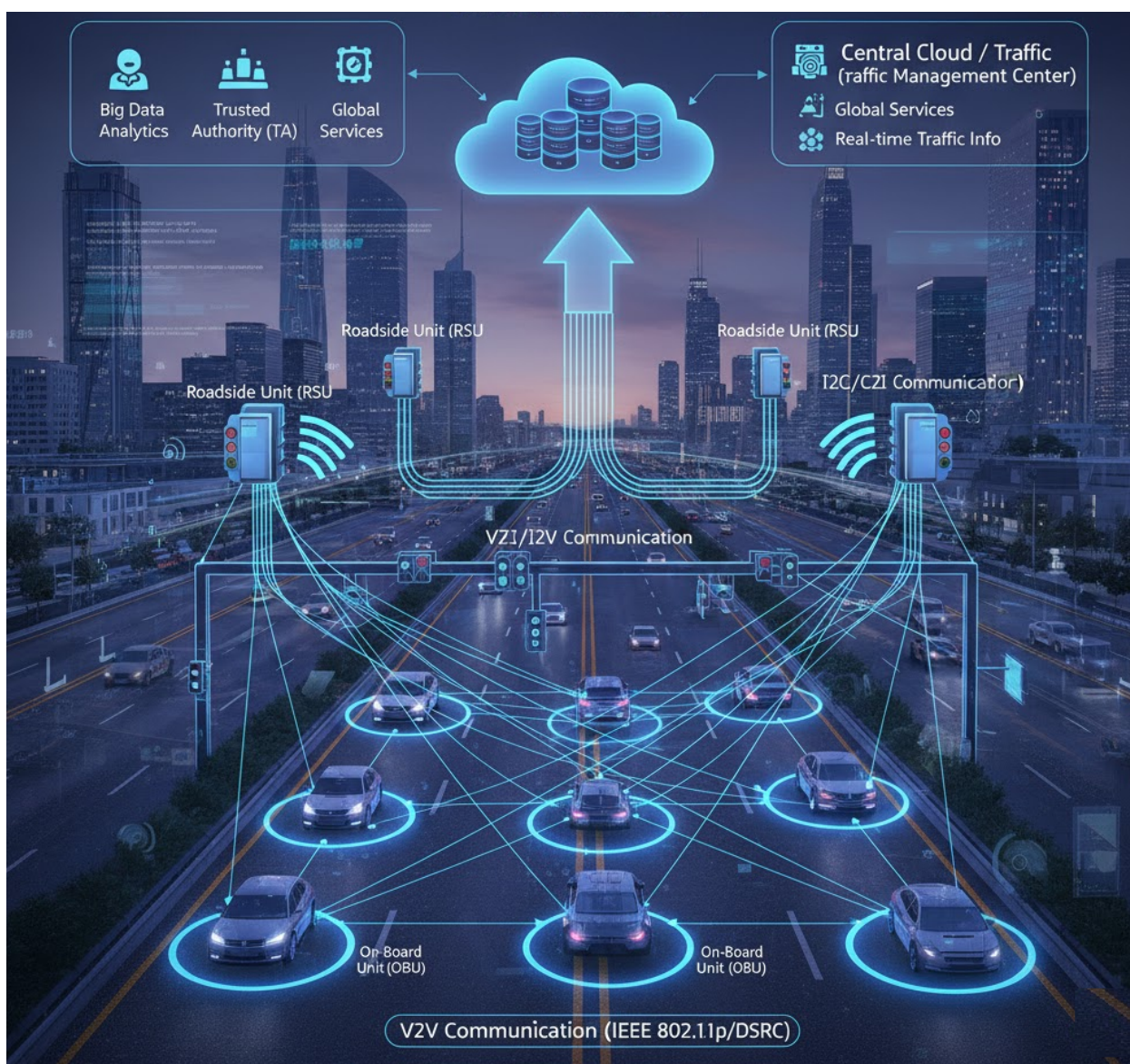


Рис. 1.1. Архітектура Vehicular Ad-hoc Network (VANET), як ключовий елемент інтелектуальних транспортних систем

Архітектура складається з трьох основних інтерактивних елементів:

А. Рівень Транспортних Засобів (Edge/Vehicle Layer)

1. Транспортні засоби (Vehicles) - автомобілі виступають як мобільні вузли.

2. Бортові блоки (On-Board Unit, OBU) - комунікаційні пристрої, вбудовані в автомобілі. Вони збирають дані (швидкість, місцезнаходження, дорожні умови) та забезпечують усі типи бездротового зв'язку (V2V, V2I).

Б. Рівень доступу (Access/Infrastructure Layer)

1. Придорожні блоки (Roadside Unit, RSU) - стаціонарні пристрої, розташовані вздовж доріг (наприклад, на світлофорах). Вони діють як точки доступу до ширшої мережі, агрегуючи дані від OBU та ретранслюючи сервіси від Центральної Хмари.

В. Рівень керування/обробки (Core/Cloud Layer)

Центральна хмара / центр керування рухом (Central Cloud / Traffic Management Center) - це вищий рівень, що відповідає за обробку великих даних. Основні функції це Big Data Analytics (аналітика великих даних), надання Global Services (глобальних сервісів), забезпечення Real-time Traffic Info (інформації про трафік у реальному часі) та управління безпекою через Trusted Authority (ТА) (довірений орган).

Ця архітектура створює динамічне, високоінтелектуальне середовище, яке використовує як мобільні, так і стаціонарні ресурси для підвищення безпеки, ефективності та зручності пересування.

1.2. Концептуалізація розподілених обчислювальних систем на базі інтернету речей

1.2.1. Еволюція та масштаби інтернету речей (IoT)

Інтернет речей (IoT) визначається як сукупність фізичних пристроїв, інтегрованих у глобальну мережу. Історично початкові пристрої IoT обмежувалися клієнтськими терміналами, такими як смартфони та планшетні

комп'ютери. Однак, технологічний прогрес у сфері мікропроцесорних технологій сприяв експоненційному зростанню категорії "розумних" пристроїв, включаючи інтелектуальні годинники, окуляри, лічильники та підключені транспортні засоби (connected vehicles). Прогнози свідчать про те, що до 2025 року кількість користувачів цих пристроїв перевищить 6 мільярдів, а загальна кількість підключених пристроїв IoT може сягнути 100 мільярдів до 2026 року.

1.2.2. Феномен крайових обчислень (Edge Computing)

Ці підключені пристрої формують периферію Інтернету, позначаючись як пристрої на краю (edge devices). Очікується, що до 2026 року обсяг даних, що генерується цим сегментом, перевищить 34.3 ексабайта щомісячно. Ці дані володіють значною інформаційною цінністю.

Внаслідок диспропорції між обсягом згенерованих даних та пропускною здатністю мережі (bandwidth gap), а також через тимчасову чутливість даних, виникає імперативна необхідність обробляти значну частину цього інформаційного потоку на краю, мінімізуючи втрати даних. Обробка, наближена до реального часу, є критичною, оскільки латентність, пов'язана з транспортуванням даних до віддалених хмарних центрів, може зробити хмарно-орієнтовану обробку неефективною або економічно недоцільною. Таким чином, виникає стратегічна потреба у розробці парадигм крайових обчислень.

Розглянемо алгоритм потоку даних Edge Computing.

1. Генерація (Generate) - IoT-пристрій генерує великий обсяг сирих даних (Raw Data) у реальному часі.
2. Передача до краю (Transmit to Edge) - дані надсилаються до найближчого Крайового Вузла.
3. Критерій затримки (Latency Check) - крайовий вузол оцінює, чи потрібна обробка в реальному часі.

- Якщо $T_{обробки} \leq T_{критичний}$ (висока чутливість до часу): → Локальна обробка (Edge Processing). Відповідь → Пристрій.

- Якщо $T_{обробки} > T_{критичний}$ (низька чутливість до часу) або потрібна глобальна аналітика → Передача до хмари (Data Aggregation & Transfer to Cloud).

4. Хмарна обробка (Cloud Processing) - виконується складний аналіз.

5. Зворотний зв'язок (Feedback) - результати аналізу (наприклад, оновлені моделі ML) відправляються назад на крайовий вузол для покращення локальної обробки.

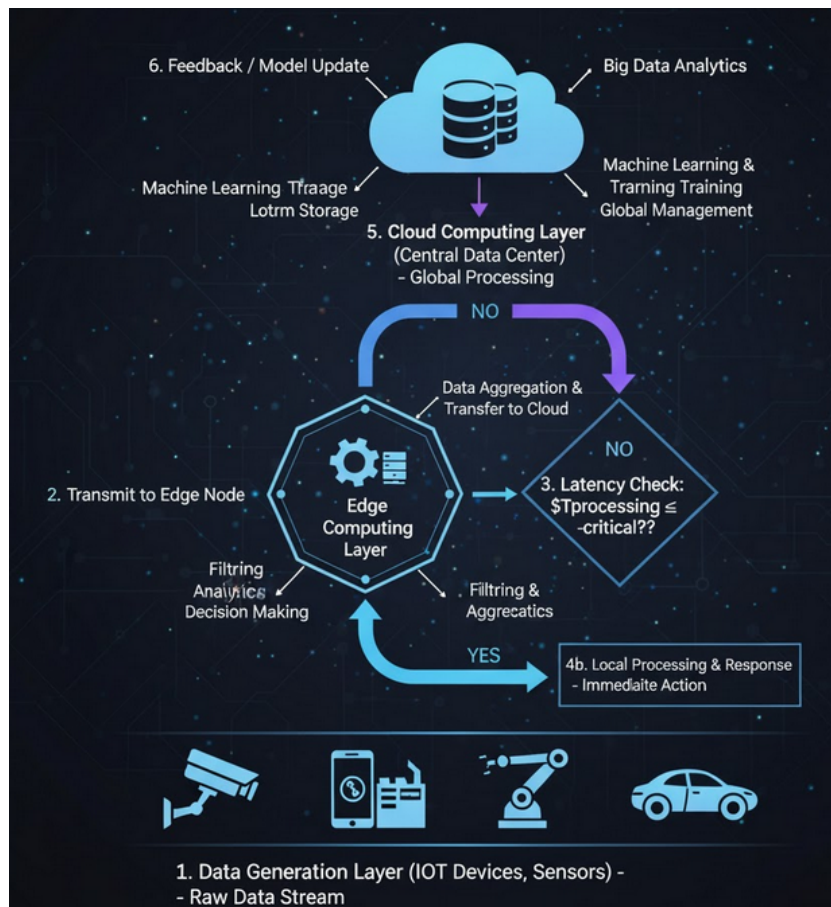


Рис. 1.2. Потік даних Edge Computing

Ключова ідея полягає в тому, що Edge Computing мінімізує обсяг даних, що передається до хмари, та скорочує час відповіді для критичних завдань. Це досягається завдяки розподілу обчислювальних функцій по всій мережі.

Пристрої на краю, які є носіями обчислювальних ресурсів та ресурсів зберігання, як правило, недовикористовуються. За оцінками, сукупна обчислювальна потужність та ємність зберігання смартфонів глобально перевищила аналогічні параметри світових серверних систем наприкінці 2017 року. Це свідчить про потенціал крайових пристроїв виконувати функції розподілених серверів.

Подальша реалізація цього потенціалу підтримується технологічними інноваціями:

- нові технології акумуляторів (наприклад, StoreDot, uBeam) покращують автономність живлення.

- Стандарти LTE Direct, WiFi Direct та WiGig сприяють збільшенню прямого зв'язку "пристрій-пристрій" (device-to-device).

- Контейнеризація (наприклад, Docker на Android) підвищує портативність додатків, вирішуючи проблеми гетерогенності та безпеки пристроїв на краю, аналогічно до ролі віртуалізації у серверному середовищі.

1.2.3 Інтелектуальний транспортний засіб як обчислювальний вузол

Особливий інтерес серед крайових пристроїв викликає розумний автомобіль (smart vehicle). Він оснащений достатніми обчислювальними та пам'ятними ресурсами не лише для керування функціями автомобіля, але й для надання додаткових сервісів оператору (ситуаційна обізнаність, комунікація, розваги). Тривають активні дослідження щодо використання сенсорних та обчислювальних ресурсів розумних автомобілів для формування динамічних сенсорних мереж (dynamic sensing networks). Хоча смартфони наразі домінують у сегменті IoT, інтелектуальні автомобілі демонструють швидку динаміку зростання.

З погляду обчислювальної інженерії, бездіяльний процесорний час є втратою ресурсу. Це стосується будь-якого обчислювального ресурсу. Сучасні автомобільні мережі (VANETs) ефективно використовують свої

ресурси в мобільному стані. Однак, коли транспортні засоби припарковані, ці ресурси переходять у бездіяльний режим.

Ключове питання: Чи можна ефективно використовувати ці бездіяльні ресурси припаркованих транспортних засобів?

Якщо припарковані автомобілі мають інтеграцію з більшою обчислювальною інфраструктурою, вони можуть контрибутувати свої ресурси в цю систему. Агрегація достатньої кількості взаємопов'язаних автомобілів створює розподілене обчислювальне середовище, здатне підтримувати виконання хмарних додатків, зокрема, обробку великих даних (Big Data).

Ця робота зосереджена на дослідженні використання припаркованих автомобілів як основи для формування інфраструктури мобільного дата-центру.

1.2.4. Методологія моделювання

Для дослідження практичності концепції буде розроблено модель, що оцінює можливість обробки великих даних за допомогою автомобільного дата-центру. Модель імітуватиме дата-центр, сформований автомобілями на парковці комерційного підприємства, яке функціонує у режимі 24/7.

Вихідні припущення моделі:

- Наявність пулу ресурсів - співробітники бізнесу працюють позмінно (наприклад, 8-годинні зміни), забезпечуючи постійний, але динамічний пул припаркованих автомобілів.

- Стимулювання ресурсу - власникам автомобілів надається стандартна послуга електричної зарядки (або інше заохочення) в обмін на використання їхніх надлишкових обчислювальних ресурсів.

Це обговорення послужить базовою лінією для цієї обчислювальної інфраструктури та основою для подальшого емпіричного дослідження використання цих невикористаних ресурсів.

1.3. Інтернет речей (IoT) в контексті розвитку транспортних мереж

Широке впровадження високошвидкісних широкосмугових комунікаційних технологій та спрощення процедур підключення пристроїв спричинило експоненційний ріст кількості мережевих пристроїв. Сучасна інфраструктура Інтернету вже не обмежується традиційними комп'ютерними системами; вона включає широкий спектр об'єктів, таких як смартфони, побутова техніка (кавоварки, пральні машини), носійні пристрої (wearables) та елементи освітлення. Аналітичні звіти підкреслюють цю тенденцію: компанія Gartner прогнозувала, що до 2025 року кількість підключених пристроїв перевищить 46 мільярдів, тоді як Cisco передбачала, що кількість пристроїв IoT досягне 100 мільярдів до того ж року.

Інтернет Речей (IoT) слід розглядати не як самоціль, а як інструментальний засіб для досягнення вищої мети — підвищення ефективності використання часу та оптимізації повсякденних процесів. Підключення забезпечує можливості для автоматизації рутинних завдань (наприклад, автоматичне замовлення продуктів холодильником на основі моніторингу запасів) та перерозподілу тимчасових ресурсів (наприклад, використання часу поїздки на роботу для виконання професійних завдань завдяки автономному управлінню розумним автомобілем). У ширшому масштабі, IoT є фундаментальною основою для реалізації концепції "розумних міст" (smart cities), метою яких є мінімізація відходів та підвищення енергетичної ефективності.

1.3.1. Роль розумних автомобілів у транспортних мережах

В основі архітектури інтелектуальних транспортних мереж знаходиться розумний автомобіль (smart vehicle). Об'єднання сенсорних ресурсів та обчислювальних потужностей розумних автомобілів є критичним для забезпечення функціональності майбутніх "розумних міст". Сукупна обчислювальна потужність (aggregate computational power) цього парку

транспортних засобів матиме значний обсяг, що дозволить обробляти колосальні масиви даних (vast volumes of data), зібраних різноманітними бортовими датчиками.

Розглянемо ключові компоненти та взаємодію в роботі Smart Vehicle (розумного автомобіля). Інтелектуальний автомобіль – це складна система, яка інтегрує передові датчики, обчислювальні потужності, комунікаційні можливості та актуатори для виконання різноманітних функцій.

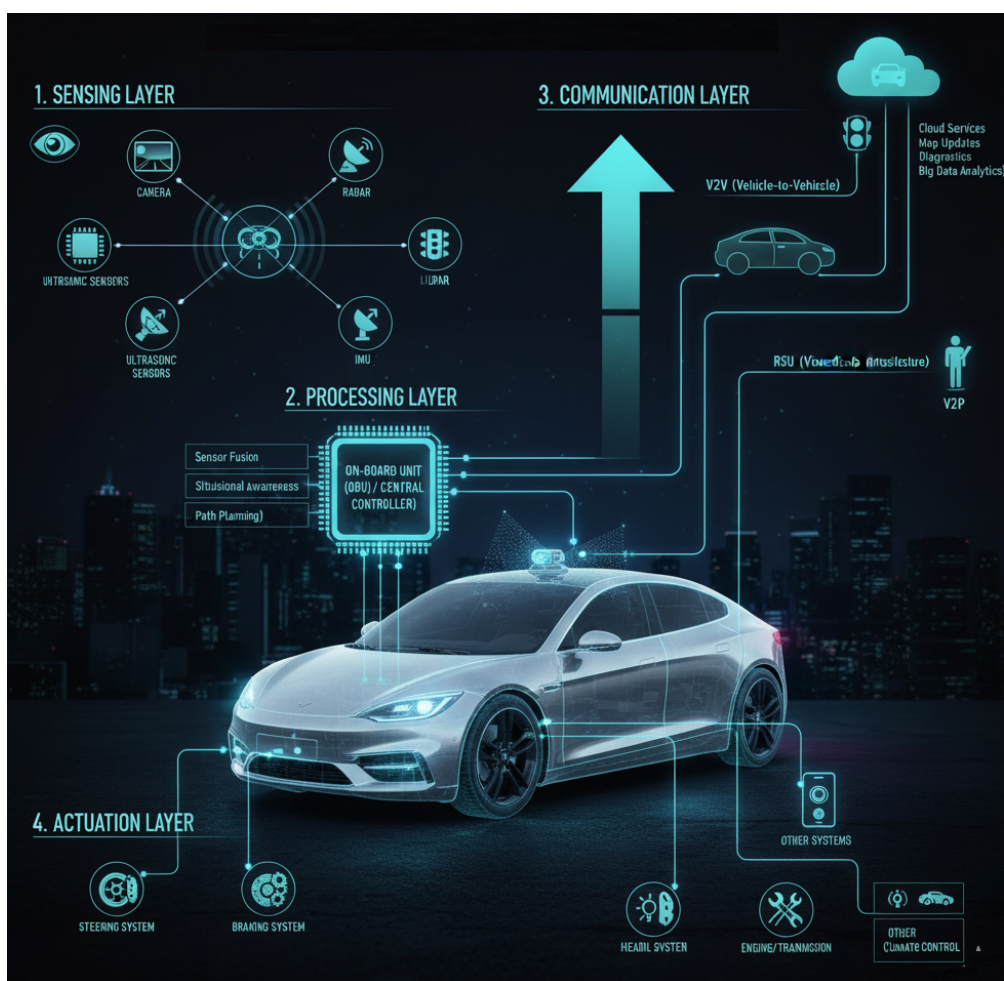


Рис. 1.3. Ключові компоненти та їх взаємодія в Smart Vehicle

1. Сенсорний Рівень (Sensing Layer)

- Камери (Cameras) - виявляють дорожні знаки, розмітку, пішоходів, інші транспортні засоби, світлофори та перешкоди. Забезпечують візуальну інформацію про оточення.

- Радари (Radar) - вимірюють відстань та швидкість об'єктів (наприклад, для адаптивного круїз-контролю, виявлення перешкод). Працюють у складних погодних умовах.

- Лідари (Lidar) - використовують лазерні промені для створення детальних 3D-карт оточення (високоточне картографування, виявлення об'єктів).

- Ультразвукові датчики (Ultrasonic Sensors) - застосовуються для паркування та виявлення об'єктів на близькій відстані.

- GPS / GNSS - визначає точне місцезнаходження автомобіля на карті.

- Інерційні вимірювальні блоки (IMU) - вимірюють орієнтацію, кутову швидкість та лінійне прискорення автомобіля.

2. Обчислювальний Рівень (Processing/Compute Layer)

- Бортовий блок (On-Board Unit, OBU) / центральний контролер автомобіля.

Функції:

- Інформація від усіх датчиків об'єднується та інтерпретується для створення повної картини оточення.

- Алгоритми (AI/ML) приймають рішення щодо подальших дій автомобіля (прискорення, гальмування, поворот, зміна смуги).

- Визначає оптимальний шлях руху автомобіля.

- Аналіз поточних умов руху, ризиків та потенційних небезпек.

- Керування бортовими системами для забезпечення комфорту та розваг.

3. Рівень Комунікації (Communication Layer)

- V2V (Vehicle-to-Vehicle): Обмін інформацією з іншими транспортними засобами.

- V2I (Vehicle-to-Infrastructure): Зв'язок з дорожньою інфраструктурою (наприклад, світлофорами).

- V2C (Vehicle-to-Cloud): Підключення до хмарних сервісів (оновлення карт, діагностика, Big Data аналітика).

- V2P (Vehicle-to-Pedestrian): Потенційний зв'язок з пішоходами.

4. Рівень Актуаторів (Actuation Layer)

- Рульове управління (Steering System) - виконання поворотів керма.

- Гальмівна Система (Braking System) - контроль прискорення/уповільнення.

- Двигун/трансмсія (Engine/Transmission) - керування швидкістю та передачами.

- Інші системи - керування фарами, двірниками, клімат-контролем, інформаційно-розважальними системами.

Розумний автомобіль не тільки взаємодіє зі своїм безпосереднім оточенням, але й інтегрується в ширшу екосистему Інтернету Речей, обмінюючись даними з іншими автомобілями, міською інфраструктурою та хмарними сервісами. Це дозволяє йому не тільки функціонувати автономно, але й робити внесок у колективну інтелектуальну транспортну систему.

1.3.2. Критичні виклики безпеки в екосистемі IoT

Незважаючи на потенційні переваги, безпека (security) є однією з найбільш важливих проблем, пов'язаних із широким впровадженням IoT. Недоліки в системах безпеки створюють вразливості для:

1. Розумних будинків.

Існують задокументовані випадки несанкціонованого доступу, які включають втручання в системи моніторингу дітей, зміну налаштувань термостатів або несанкціоноване відмикання дверей.

2. Транспортних засобів.

Оскільки бортові процесори автомобілів контролюють все більше критичних систем, вони стають привабливою ціллю для кібератак. Були продемонстровані випадки, коли зловмисники отримували повний дистанційний контроль над системами автомобіля, включаючи кермове управління, гальма та двигун, що створює безпосередню загрозу безпеці.

З огляду на експансію IoT, посилення пильності та розробка надійних механізмів захисту для підключених пристроїв є імперативною вимогою для забезпечення функціональної цілісності та безпеки користувачів.

1.4. Науковий аналіз зростання великих даних та розподілені обчислювальні стратегії

Сучасна життєдіяльність характеризується інтенсивною генерацією даних (data deluge), обсяг яких експоненційно зростає, головним чином, завдяки розширенню Інтернету Речей (IoT). Соціальні мережі ілюструють цей феномен: користувачі щоденно створюють значні обсяги високоякісного фото- та відеоконтенту. IoT охоплює широкий спектр пристроїв, що здійснюють збір даних: від побутової техніки та носійних пристроїв до інтелектуальних транспортних засобів та сенсорних мереж.

Прогнози демонструють стрімке збільшення кількості підключених пристроїв: оцінюється, що поточна кількість перевищує 50 мільярдів одиниць і, за очікуваннями, зросте до 75 мільярдів до 2025 року. Крім того, прогнозувалося, що щомісячний обсяг даних, генерованих крайовими пристроями (смартфони, носійні пристрої, автомобілі), перевищить 44.3 ексабайта до 2026 року. Це зростання надає потенційну можливість для паралельної обробки створюваних великих наборів даних (Big Datasets).

1.4.1. Вимоги до обробки великих даних

Ефективна обробка великих даних є критичною для підтримки функціональності численних сучасних додатків:

Електронна комерція (E-commerce). У цьому сегменті першочергове значення має досвід користувача (User Experience, UX). Управління даними, що стосуються пошукових запитів, вмісту кошиків, уподобань, історії покупок та повернень клієнтів, вимагає високоефективного зберігання та

швидкого вилучення інформації. Будь-яка латентність (latency) у наданні інформації може призвести до незадоволення клієнта та його втрати.

Композитні послуги та геопросторові запити. Складні запити, наприклад, визначення оптимального маршруту до місця призначення з інтеграцією інформації про супутні послуги (готелі, ресторани), вимагають обробки в режимі, близькому до реального часу (near real-time processing). Це передбачає інтеграцію поточних даних датчиків (для умов руху) з обробленими статичними даними (для інфраструктури). Необхідно забезпечити своєчасну доставку агрегованих результатів клієнту, незважаючи на потенційну відмовостійкість (fault tolerance) мережі та розподіл даних у різних географічних локаціях.

Обробка великих даних, яка передбачає маніпуляції з терабайтами або петабайтами інформації, робить традиційні підходи неефективними. Традиційний метод, що базується на обробці єдиного набору даних однією програмою на одному комп'ютері, призводить до надмірно тривалого часу обробки. Це може зробити отримані результати неактуальними на момент завершення аналізу.

Основною перешкодою в обробці великих даних є затримка. Враховуючи експоненційний ріст обсягу даних, просте оновлення апаратного забезпечення (hardware upgrades) є недостатнім для забезпечення необхідної продуктивності. Це вимагає переходу до нових стратегій обробки даних.

1.4.2. Парадигма розподілених обчислень (Distributed Computing). Принцип MapReduce

Ключ до вирішення проблеми полягає не стільки у зміні апаратного забезпечення, скільки у зміні методу обробки даних. Сучасні додатки для великих даних часто передбачають складну багатозадачну обробку.

Рішення, такі як MapReduce від Google та Hadoop від Apache, є основними інструментами для розподіленої обробки великих даних. Принцип MapReduce реалізує обробку у двох послідовних стадіях:

Етап Map: до кожного логічного вхідного запису застосовується користувацька функція, яка генерує проміжний результат у формі пар «ключ-значення».

Етап Reduce: збирає всі проміжні пари «ключ-значення» з етапу Map та агрегує/комбінує їх за допомогою іншої користувацької функції.

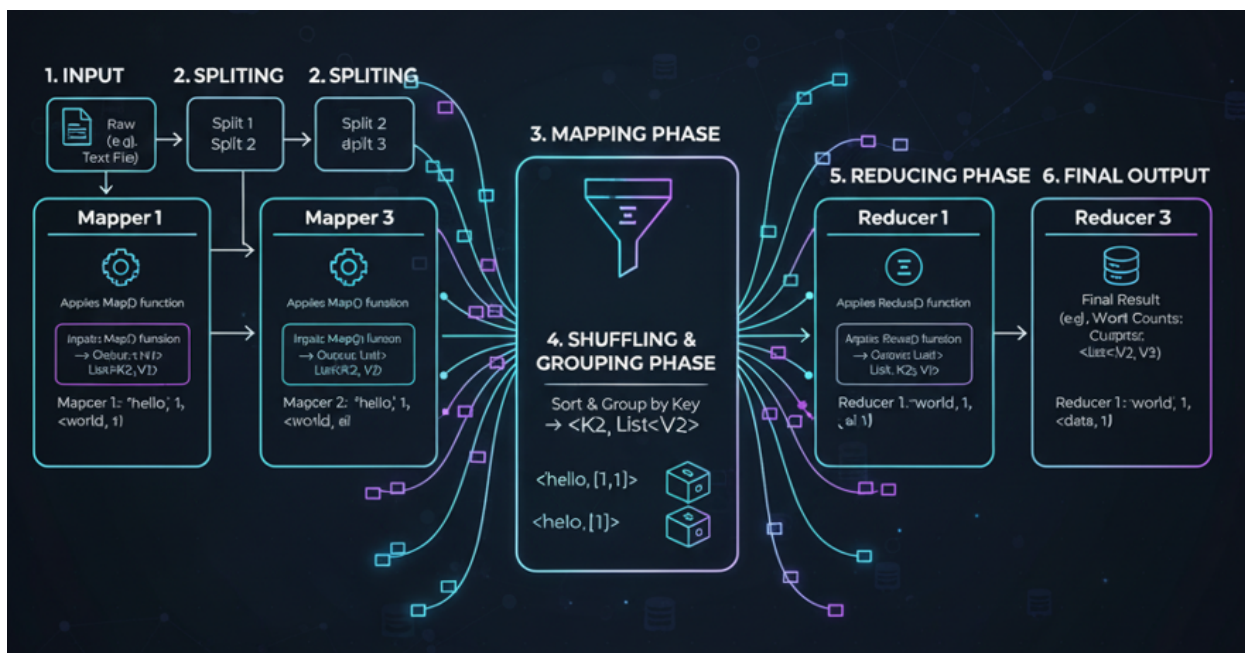


Рис. 1.4. Графічна інтерпретація алгоритму MapReduce

MapReduce — це потужна модель, що використовує підхід "розділяй і володарюй" для паралельної обробки великих наборів даних на кластері розподілених машин. Весь процес поділяється на кілька ключових фаз, де дані обробляються виключно у форматі пар "ключ-значення" (<Key, Value>).

Цей метод використовує принцип розподілених обчислень. Застосування множинних обчислювальних вузлів (nodes) для паралельного виконання етапів Map і Reduce дозволяє досягти значного підвищення продуктивності та забезпечити обробку, наближену до реального часу.

Таким чином, MapReduce забезпечує горизонтальну масштабованість і відмовостійкість, оскільки кожен етап Map може виконуватися незалежно, а збій одного вузла не призводить до втрати даних.

Опис потоку даних та фаз алгоритму MapReduce

Фаза	Дія	Формат Даних (Вхід → Вихід)	Пояснення
1. Вхід (Input)	Зчитування сирих даних.	(Сирі дані)	Великий набір даних, що зберігається у розподіленій файловій системі (наприклад, HDFS).
2. Розділення (Splitting)	Розбиття вхідних даних на незалежні, менші блоки (Split).	(Сирі дані) → ((Byte Offset, Text Block))	Забезпечує паралельну обробку: кожен блок призначається окремому Mapper-процесу.
3. Мапування (Mapping)	Мапер (Mapper) застосовує користувацьку функцію Map() до кожного елемента в блоці.	$\langle K1, V1 \rangle \rightarrow \text{List}(\langle K2, V2 \rangle)$	Перетворює вхідні пари на проміжні пари "ключ-значення" для подальшої агрегації.
4. Перемішування (Shuffling)	Проміжний етап, який керується фреймворком.	$\text{List}(\langle K2, V2 \rangle) \rightarrow \langle K2, \text{List}(V2) \rangle$	Сортування (Sorting) та Групування (Grouping): усі проміжні пари з однаковим ключем (K2) збираються разом і надсилаються до одного Редьюсера.
5. Редукування (Reducing)	Редьюсер (Reducer) застосовує користувацьку функцію Reduce() до групи значень.	$\langle K2, \text{List}(V2) \rangle \rightarrow \text{List}(\langle K3, V3 \rangle)$	Виконує фінальну агрегацію, підсумовування або об'єднання даних для кожного унікального ключа (наприклад, підрахунок частоти слова).
6. Вихід (Final Output)	Збереження кінцевого результату.	$\text{List}(\langle K3, V3 \rangle)$	Оброблені дані записуються у розподілену файлову систему.

У деяких реалізаціях, як-от Hadoop, між фазами Mapping та Shuffling може використовуватися опціональна фаза Combiner (міні-редьюсер), яка локально агрегує проміжні результати на тому ж вузлі, де був виконаний Map, щоб зменшити обсяг даних для передачі через мережу (фаза Shuffling).

Висновки до розділу

У першому розділі проведено ґрунтовний аналіз предметної області застосування транспортних мереж у контексті розподілених обчислень і великих даних. Встановлено, що Vehicular Ad-hoc Networks (VANETs) є ключовою технологією для побудови децентралізованих обчислювальних середовищ, здатних підтримувати мобільну взаємодію транспортних засобів у реальному часі. Визначено, що інтеграція IoT і Edge Computing створює новий рівень інтелекту в транспортних системах, забезпечуючи ефективне використання обчислювальних ресурсів на периферії мережі. Досліджено архітектурні підходи до побудови дата-центрів і методологію моделювання обчислювальних процесів у транспортних мережах. Показано, що еволюція IoT і зростання обсягів даних зумовлюють необхідність переходу від централізованих до розподілених моделей обробки інформації. Таким чином, у межах розділу сформовано наукове підґрунтя для подальшої розробки методів реалізації транспортних мереж, орієнтованих на великі дані.

РОЗДІЛ 2. МЕТОДИ ТА КОНЦЕПЦІЇ РЕАЛІЗАЦІЇ ТРАНСПОРТНИХ МЕРЕЖ НА ОСНОВІ КОНЦЕПЦІЇ ВЕЛИКИХ ДАНИХ

2.1. Консолідація ресурсів та моделі обслуговування в парадигмі хмарних обчислень

Хмарні обчислення (Cloud Computing) є домінуючою рушійною силою у сфері інформаційних технологій, визначаючи методи розгортання додатків та управління обчислювальними ресурсами. Сутність концепції полягає в консолідації розрізаних обчислювальних потужностей у великомасштабні, централізовані об'єкти.

Переваги консолідації:

- Мінімізація витрат на інфраструктуру: Дозволяє досягти значної економії від масштабу.
- Спрощення адміністрування: Централізоване управління підвищує ефективність операцій.

Ключовим технологічним елементом є віртуалізація, яка забезпечує абстрагування обчислювальних ресурсів від базового фізичного апаратного забезпечення. Це сприяє ефективному використанню ресурсів та забезпечує високу обслуговуваність і надійність. У разі потреби технічного обслуговування або ремонту фізичних серверів віртуалізовані ресурси можуть бути оперативно мігровані на інші фізичні вузли.

2.1.1. Моделі хмарного обслуговування

Хмарні обчислення визначаються як надання обчислювальних послуг за вимогою, включаючи сервери, сховища, бази даних, мережеві компоненти, програмне забезпечення та аналітичні інструменти, через мережу (зазвичай Інтернет). Модель оплати "за фактичне використання" (pay-as-you-go) дозволяє клієнтам знижувати капітальні та операційні витрати, оскільки вони

уникають накладних витрат, пов'язаних з володінням фізичними ресурсами, об'єктами та утриманням спеціалізованого персоналу.

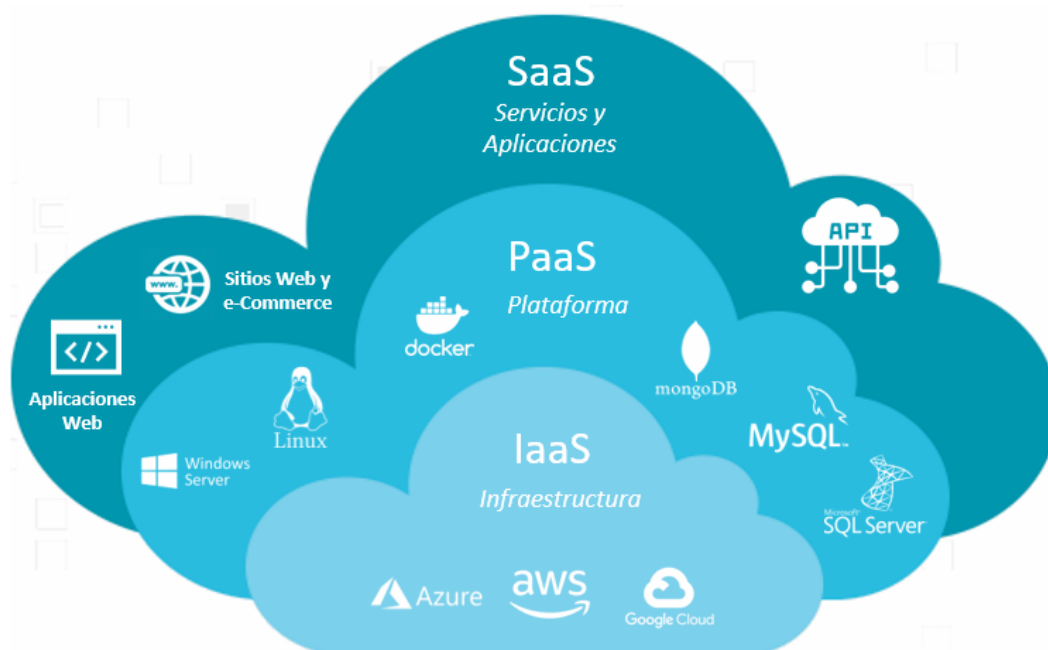


Рис. 2.1. Моделі хмарного обслуговування

Загальноприйнятими є три основні моделі хмарного обслуговування:

А. Інфраструктура як Послуга (Infrastructure as a Service, IaaS)

IaaS пропонує клієнтам базові обчислювальні ресурси, такі як обчислювальні потужності, оперативна пам'ять (RAM) та сховище, у вигляді віртуальних серверів. Клієнт визначає необхідні параметри та отримує віртуалізовані ресурси, зберігаючи контроль над операційною системою та додатками.

Приклад: Amazon Web Services (AWS) надає обчислювальні ресурси через Elastic Compute Cloud (EC2) та сховище через Simple Storage Service (S3) та Elastic Block Store (EBS).

Б. Платформа як Послуга (Platform as a Service, PaaS)

PaaS надає клієнтам платформу для розробки та розгортання додатків. Це звільняє клієнта від необхідності управління та підтримки базової інфраструктури (операційні системи, бази даних, проміжне програмне забезпечення).

Приклад: Веб-хостинг, де клієнти можуть розробляти та впроваджувати веб-сайти, використовуючи лише веб-браузер. Прикладами є Google AppEngine та Microsoft Azure.

В. Програмне забезпечення як послуга (Software as a Service, SaaS)

SaaS пропонує програмне забезпечення за моделлю підписки (pay-per-use). Користувачі отримують доступ до готового програмного забезпечення через Інтернет, усуваючи необхідність його встановлення та управління. Ця модель особливо вигідна для клієнтів, які не можуть дозволити собі високу вартість ліцензування дорогого програмного забезпечення та супутніх інфраструктурних ресурсів.

Приклад: Послуги, що надаються через Google AppEngine (в контексті кінцевого користування додатками).

2.1.2. Дата-центр як основа хмарних обчислень

Дата-центр (Data Center) є фізичною основою хмарних обчислень. Це консолідована сукупність апаратних компонентів, що включає сервери, маршрутизатори, мережеві комутатори та системи зберігання даних (дискові бібліотеки). Розміри дата-центрів можуть варіюватися від невеликих приміщень до великих, спеціалізованих складських комплексів.

Дата-центри забезпечують ключові елементи, без яких хмарні обчислення були б неможливими:

- Обчислювальні потужності: Тисячі серверів надають процесорний час та оперативну пам'ять для виконання мільйонів операцій одночасно.

- Зберігання даних: Величезні масиви систем зберігання даних (СЗД) надійно розміщують інформацію користувачів та компаній.

- Мережева інфраструктура: Складне мережеве обладнання забезпечує миттєвий зв'язок між серверами та швидкий доступ до даних через інтернет.

- Безперебійність та надійність: Це одна з найважливіших функцій. Дата-центри проєктуються так, щоб працювати 24/7 без збоїв.

Надійність дата-центрів класифікують за міжнародним стандартом Tier. Більшість провідних хмарних операторів використовують центри рівня Tier III (до 1.6 години простою на рік) або Tier IV (до 26 хвилин простою на рік), що гарантує надзвичайно високу доступність сервісів.

Отже, хоча ми звикли говорити про "хмару" як про віртуальний простір, її робота цілком залежить від реальної, фізичної та високотехнологічної інфраструктури дата-центрів, які є її надійним фундаментом.

2.2. Огляд досліджень Vehicular Cloud Computing (VCC) від концепції до архітектури дата-центру

2.2.1. Концепція автомобільних хмар (Vehicular Clouds)

Концепція автомобільних хмар (Vehicular Cloud, VC), яка передбачає використання кластерів транспортних засобів як динамічного обчислювального середовища, була вперше представлена в роботах [4] та [5]. Ці дослідження заклали основи для розуміння потенційних можливостей та конфігурацій VC, підтверджуючи її життєздатність при застосуванні існуючих технологій.

Обчислювальна хмара транспортних засобів (Vehicular Cloud Computing, VCC) являє собою парадигму, що інтегрує транспортні засоби в єдину обчислювальну екосистему. Вона використовує ресурси автомобілів (обчислювальну потужність, сховище даних, сенсори) для надання різноманітних сервісів як водіям, так і централізованим системам керування. Архітектура VCC є ієрархічною та, як правило, складається з трьох фундаментальних рівнів.

1. Рівень транспортних засобів (Vehicular Layer)

Це базовий рівень архітектури, що складається з рухомих вузлів — транспортних засобів. Кожен транспортний засіб розглядається як мобільний обчислювальний ресурс, оснащений бортовим блоком (On-Board Unit, OBU)

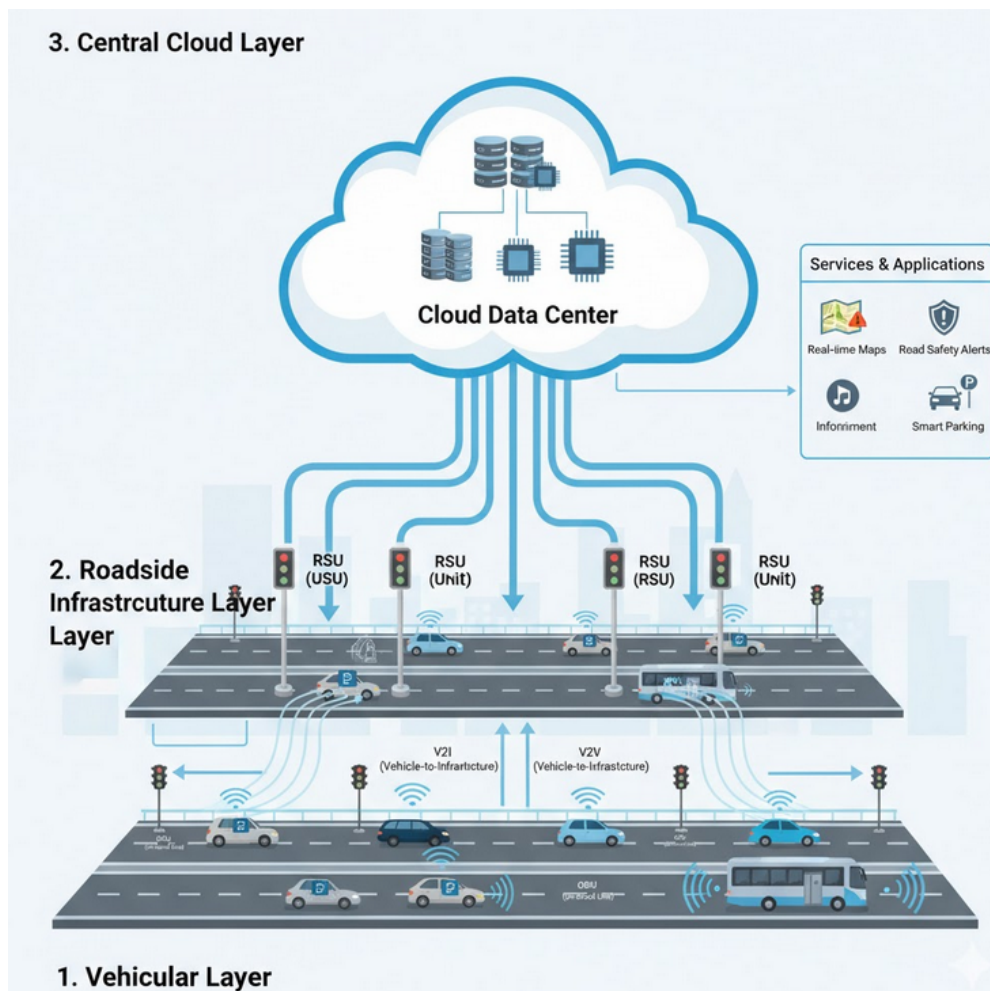


Рис. 2.2. Тривірнева архітектура Vehicular Cloud

2. Рівень Придорожньої Інфраструктури (Roadside Infrastructure Layer)

Цей проміжний рівень функціонує як граничний або "туманний" обчислювальний шар (Edge/Fog Computing Layer). Його ключовими елементами є придорожні блоки (Roadside Units, RSU), які встановлюються вздовж доріг.

3. Рівень центральної хмари (Central Cloud Layer)

Верхній рівень архітектури — це традиційна високопродуктивна хмарна інфраструктура, що базується на центрах обробки даних (Data Centers).

Функціонування системи забезпечується безперервною взаємодією між рівнями. Дані, згенеровані на Vehicular Layer, передаються через V2I з'єднання до RSU. На Infrastructure Layer відбувається їх попередня обробка,

після чого релевантна інформація надсилається до Central Cloud для глибокого аналізу. Результати обробки та сервісні команди повертаються тим же шляхом до транспортних засобів, забезпечуючи їх актуальною інформацією для підвищення безпеки та ефективності руху.

Таким чином, представлена трирівнева архітектура дозволяє ефективно розподілити обчислювальне навантаження, забезпечуючи низьку затримку для критично важливих завдань на нижніх рівнях та високу обчислювальну потужність для глобального аналізу на верхньому рівні

2.2.2. Моделювання доступності ресурсів

Подальші дослідження зосередилися на кількісній оцінці доступності обчислювальних ресурсів у таких динамічних середовищах. В роботі [7] розробили стохастичну модель для прогнозування зайнятості парковки на прикладі великого аеропорту. Вони вивели розподіл ймовірності зайнятості як функцію часу прибуття та відправлення транспортних засобів, підтверджуючи модель емпіричними даними. Це дослідження є важливим для оцінки пулу ресурсів в інфраструктурі, сформованій припаркованими автомобілями.

Кілька робіт досліджували архітектурні моделі та потенційні сервіси VC:

- Модель Master-Provider (CaaS та SaaS) [8] деталізували архітектуру майстер-провайдер, де певні автомобілі виконують роль контролерів (майстрів), що координують дії інших автомобілів-провайдерів (робітників).

- У ролі обчислення як послуга (CaaS) майстер маршрутизує обчислювальні запити від клієнтів до найбільш доступних автомобілів.

- У ролі зберігання як послуга (SaaS) майстер оптимізує розміщення та вилучення даних серед доступних вузлів.

Інтеграція з хмарою (VANET-Cloud Integration) в [9] запропонували гібридну мережу, що поєднує автомобільні спеціалізовані мережі (VANET) з традиційним хмарним середовищем. Дорожні шлюзові термінали (GT)

виступають як критичні посередники для зв'язку між автомобільною мережею та наземною хмарною інфраструктурою.

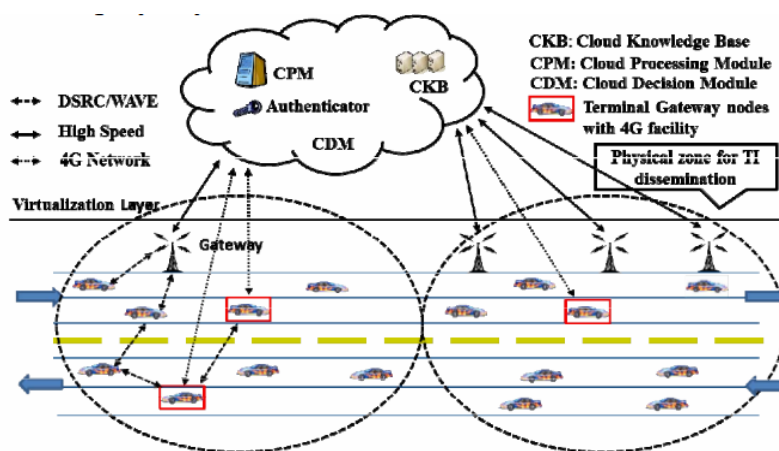


Рис. 2.3. Модель гібридної хмари

На рисунку 2.3 зображено модель мережі, де архітектуру поділяють на дві мережі (а саме VANET та хмарні обчислення (ХО)), які з'єднані через шлюзові термінали (GT). Архітектура VANET складається з традиційних транспортних вузлів, придорожніх блоків (RSU) та органів управління/відкликання повноважень. Транспортні засоби, що рухаються по дорозі, виступають одночасно і як виробники (надаючи загальну, недеталізовану інформацію хмарі), і як споживачі (підписуючись на отримання детальної інформації про дорожній рух від хмари).

Придорожні блоки (RSU) слугують шлюзовими терміналами (GT) між транспортними засобами та хмарною інфраструктурою. Ми припускаємо, що частина автомобілів на дорозі матиме доступ до інтернету 4G/5G, який можна буде використати як вторинний шлюзовий термінал до хмари в тих регіонах, де придорожні блоки (RSU) ще не розгорнуто.

2.2.3. IoT-орієнтовані сервіси та обробка великих даних

В дослідженні [10] запропонували ряд сервісів, які можуть бути реалізовані через IoT-орієнтовані автомобільні хмарні дані (IoT-oriented Vehicular Data Clouds). Ці сервіси включають прогнозування безпеки на

дорогах, оптимізацію транспортних потоків (зменшення заторів) та системи рекомендацій щодо паркування. Вони підкреслили, що для успішної реалізації цих послуг VC-архітектури повинні одночасно задовольняти вимоги ефективності, масштабованості, безпеки та надійності. Вони дійшли висновку, що існуючі алгоритми не можуть повністю задовольнити всі ці критерії комплексно.

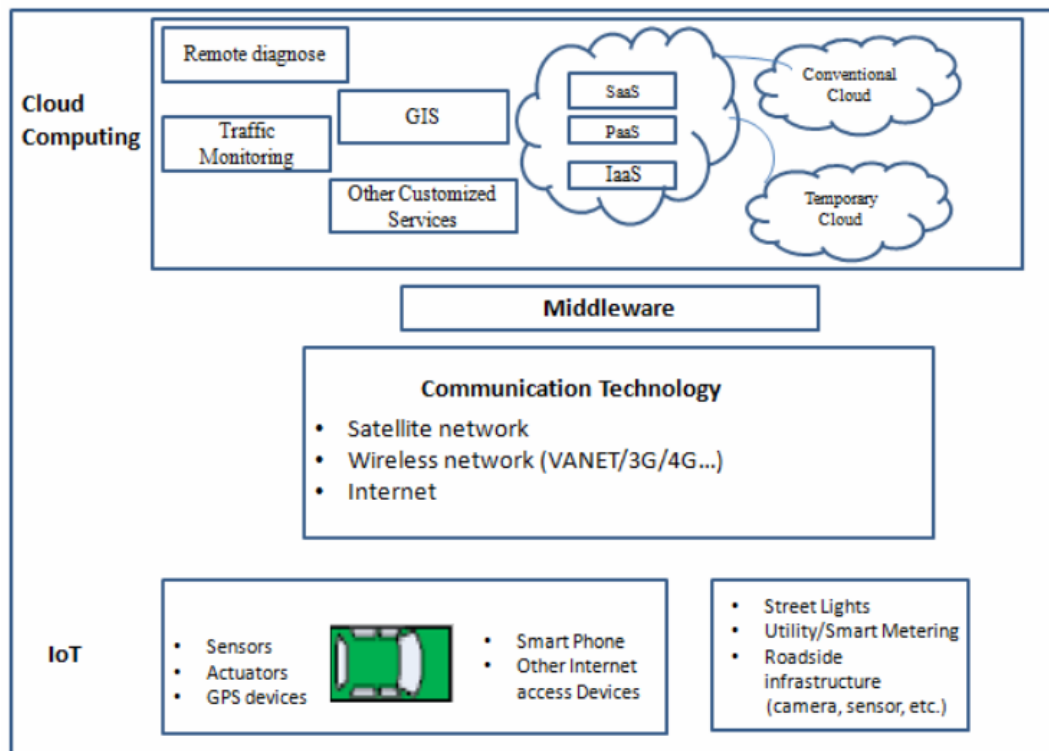


Рис. 2.4. Багаторівнева архітектура хмарної платформи транспортних даних на основі Інтернету речей

На рисунку 2.4 показано багаторівневу архітектуру хмарної платформи транспортних даних на основі Інтернету речей (IoT). Завдяки інтеграції різноманітних пристроїв (таких як сенсори, виконавчі механізми, контролери, GPS-пристрої, мобільні телефони та інше обладнання для доступу до Інтернету) та застосуванню мережевих технологій (бездротова сенсорна мережа, стільникова мережа, супутникова мережа та інші), хмарних обчислень, IoT та проміжного програмного забезпечення (middleware), ця платформа підтримує механізми зв'язку V2V (Vehicle-to-Vehicle) та V2I

(Vehicle-to-Infrastructure) і здатна збирати та обмінюватися даними між водіями, транспортними засобами та придорожньою інфраструктурою, такою як камери і вуличні ліхтарі.

Архітектура (рис 2.4) складається з трьох основних рівнів, розділених проміжним програмним забезпеченням (Middleware):

1. Рівень Інтернету Речей (IoT Layer)

Це базовий, фізичний рівень, що відповідає за збір даних та взаємодію з фізичним світом. Він охоплює всі пристрої та об'єкти, які генерують дані.

- Транспортні засоби: самі автомобілі, оснащені сенсорами, актуаторами та GPS-пристроями, виступають як мобільні вузли IoT.
- Персональні Пристрої: смартфони та інші пристрої доступу до Інтернету, які можуть бути в автомобілі.
- Дорожня інфраструктура: стаціонарні елементи, такі як вуличні ліхтарі, комунальні/розумні лічильники та придорожня інфраструктура.

2. Рівень комунікаційних технологій (Communication Technology Layer)

Цей рівень забезпечує передачу даних від рівня IoT до хмарної платформи. Це включає різноманітні мережеві технології:

- Супутникова мережа (Satellite network): Використовується для глобального покриття або у віддалених районах.
- Бездротова мережа (Wireless network): Включає VANET (Vehicular Ad hoc Network), а також комерційні стільникові мережі (3G/4G/5G) для зв'язку між транспортними засобами, інфраструктурою та хмарою.
- Інтернет (Internet): Основа для маршрутизації та обміну даними.

Між комунікаційним рівнем та хмарним рівнем розташоване проміжне програмне забезпечення (Middleware), яке відповідає за управління зв'язком, агрегацію даних, форматування та забезпечення безпеки перед їхньою передачею до хмари.

3. Рівень хмарних обчислень (Cloud Computing Layer)

Це верхній, логічний рівень, що відповідає за обробку, зберігання та аналіз даних, а також за надання кінцевих послуг.

Моделі розгортання хмари:

- Конвенційна хмара (Conventional Cloud): Стаціонарні, великі дата-центри, що надають основні послуги.

- Тимчасова хмара (Temporary Cloud): динамічно сформовані хмари, такі як автомобільні хмари (Vehicular Clouds) або Edge Clouds, що утворюються з групи сусідніх транспортних засобів для локальної та швидкої обробки даних.

Моделі Послуг (XaaS):

Хмарний рівень надає послуги через стандартні моделі: SaaS, PaaS, IaaS.

На основі оброблених даних платформа надає низку прикладних послуг:

- Віддалена діагностика (Remote diagnose): Моніторинг стану транспортних засобів.

- Моніторинг трафіку (Traffic Monitoring): Збір та аналіз даних про дорожній рух.

- GIS (Geographic Information System): Послуги, пов'язані з географічним розташуванням та картографуванням.

- Інші налаштовувані послуги: Будь-які додаткові послуги, що можуть бути реалізовані на базі цієї архітектури (наприклад, рекомендації щодо паркування, попередження про небезпеку).

Метою цієї платформи є надання клієнтам економічних, безпечних послуг на вимогу в режимі реального часу через пов'язані хмари, що включають традиційну хмару та тимчасову хмару (транспортну хмару) [3]. Традиційна хмара складається з віртуалізованих комп'ютерів і надає зацікавленим клієнтам послуги за моделями SaaS, PaaS та IaaS. Наприклад, у традиційній хмарі можуть бути розміщені сервіси управління хмарою та багато додатків для адміністрування дорожнього руху.

Тимчасова хмара зазвичай формується на вимогу і складається з недовикористаних обчислювальних, мережевих ресурсів та сховищ транспортних засобів. Вона призначена для розширення традиційної хмари з метою збільшення загальних обчислювальних потужностей, можливостей

обробки та зберігання даних усієї хмари. Тимчасова хмара підтримує комбінацію SaaS, PaaS та IaaS і в основному розміщує високонамічні транспортні додатки, з роботою яких у традиційних хмарах можуть виникати проблеми. Наприклад, додатки, пов'язані з дорожнім рухом, та додатки для інтелектуального паркування добре підходять для тимчасової хмари. Тимчасовій хмарі часто необхідно взаємодіяти з традиційними хмарами, і між цими двома хмарами відбувається частий обмін даними та послугами.

На основі багаторівневої архітектури, показаної на рисунку 2.4, різноманітні пристрої, пов'язані з IoT, мережеві та комунікаційні технології, а також хмарні сервіси на різних рівнях можуть бути інтегровані для обміну інформацією, спільного використання ресурсів та взаємодії у хмарах.

В роботі [11] сфокусувалися на дослідженні VC, сформованої припаркованими автомобілями в умовах бізнесу середнього розміру. Їхнє дослідження виявило, що існуючі бездротові технології є недостатніми для ефективної підтримки додатків обробки великих даних (Big Data) на автомобільній хмарі. Як наслідок, вони досліджували техніки міграції (migration techniques) як засіб підвищення надійності обробки великих даних. Їхня модель, що передбачає постійну зайнятість більше 2000 паркомісць розумними автомобілями, послугувала основою для моделі, використаної в даній роботі.

2.3. Концептуальна архітектура та моделювання автомобільного дата-центру

2.3.1. Концепція автомобільного дата-центру (VDC)

Автомобільний дата-центр (Vehicular Data Center, VDC) є формою розподілених обчислень, яка функціонально подібна до традиційних дата-центрів, що підтримують хмарні обчислення. Ключова архітектурна відмінність полягає у фізичному розміщенні обчислювальних ресурсів: традиційні сервери в стійках замінюються розподіленими фізичними

серверами, інтегрованими в транспортні засоби, припарковані на великій території. Дане дослідження зосереджується на моделюванні великих парковок, де автомобілі залишаються протягом тривалих періодів часу, як це типово для великих аеропортів або цілодобових (24/7) середніх та великих підприємств.

Модель імітує VDC, розгорнутий на парковці середнього підприємства, що працює цілодобово (24/7), з наступними параметрами:

1. Робочий режим: підприємство наймає 7680 співробітників, які працюють у восьмигодинних змінних режимах.

2. Динаміка ресурсів: на початку кожної години 320 співробітників завершують зміну і виїжджають, а 320 нових співробітників прибувають, підтримуючи постійну зайнятість.

3. Парковка: загальна місткість становить 2560 місць. Автомобілі вибирають місця випадково. Парковка залишається повністю заповненою, за винятком короткого проміжку часу під час зміни автомобілів.

4. Стимулювання ресурсів: автомобілі підключаються до стандартної електричної розетки для зарядки в обмін на використання їхніх обчислювальних ресурсів.

5. Комунікація: для забезпечення обробки даних використовується дротове з'єднання до локальних точок доступу.

Основний високорівневий виклик для реалізації VDC полягає в підтримці високої доступності та надійності (Reliability) у динамічному середовищі, де фізичні сервери (автомобілі) можуть залишати мережу.

2.3.2. Архітектура обчислювальних ресурсів (віртуалізація)

VDC пропонує клієнтам обчислювальні ресурси у формі віртуальних машин (VM).

Хостинг VM: Припускається, що всі автомобілі мають попередньо завантажений монітор віртуальних машин (VMM), який відображає VM на ресурси автомобіля.

Ресурси автомобіля: кожен автомобіль має здатність хостувати кілька VM та достатній дисковий простір для розміщення образу VM та оброблюваних даних.

Специфікація VM: розмір кожної віртуальної машини припускається рівномірно 1 ГБ.

2.3.3. Характеристики завдань обробки великих даних

Клієнти VDC виконують завдання, використовуючи парадигму MapReduce, з наступними характеристиками:

1. Тривалість завдання. Час, необхідний для виконання завдання (без урахування накладних витрат), розподілений рівномірно між 2 та 24 годинами.

2. Вхідні дані - кожне завдання вимагає 2 ГБ сирих вхідних даних.

3. Вихідні дані - обсяг кінцевих вихідних даних розподілений рівномірно між 0.5 ГБ та 2 ГБ.

4. Проміжні дані - завдання MapReduce генерує обсяг проміжних даних (наприкінці етапу Map), що дорівнює обсягу кінцевих вихідних даних (наприкінці етапу Reduce).

Мережа, що з'єднує автомобілі, організована у вигляді ієрархічної деревоподібної структури для забезпечення структурованої комунікації (рис. 2.5).

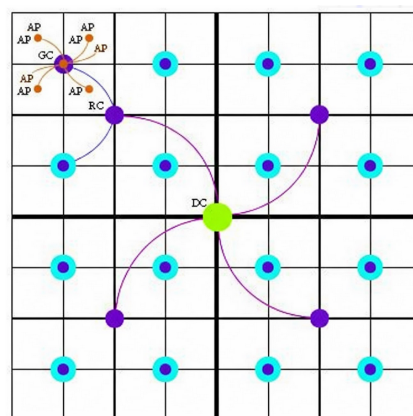


Рис. 2.5. Модель мережі

Топологія мережі VDC

Рівень	Компонент	Роль / Функція	Пропускна здатність (Uplink)
0 (Корінь)	Контролер Дата-Центру (DC)	Головний комутатор.	N/A
1	Контролер Регіону (RC)	4 дочірні елементи DC (Комутатори).	40 Гбіт/с (DC ↔ RC)
2	Контролер Групи (GC)	4 дочірні елементи RC (Комутатори).	10 Гбіт/с (RC ↔ GC)
3	Точка Доступу (AP)	4 дочірні елементи GC (Комутатори).	10 Гбіт/с (GC ↔ AP)
4 (Периферія)	Автомобілі (Сервери)	Кластер із 40 паркомісць, підключених до однієї AP.	1 Гбіт/с (AP ↔ Автомобіль)

Загальна кількість точок доступу (AP) становить:

Кількість AP = Кількість RC × Кількість GC/RC × Кількість AP/GC = 4×4×4 = 64 AP.

Загальна місткість парковки становить: 64 AP×40 місць/AP=2560 місць, що відповідає місткості парковки. Комунікація автомобілів у кластері відбувається виключно через призначену AP.

Висновки до розділу

У другому розділі розглянуто методи та концепції реалізації транспортних мереж на основі парадигми великих даних. Проаналізовано моделі хмарного обслуговування (IaaS, PaaS, SaaS) та визначено їхню доцільність для розподілених автомобільних систем. Розроблено концепцію Vehicular Cloud Computing (VCC), у якій транспортні засоби виступають як динамічні вузли хмарної інфраструктури, що дозволяє створювати автомобільні дата-центри (VDC) з розподіленими обчислювальними

можливостями. Досліджено процеси моделювання доступності ресурсів і оптимізації навантаження у таких середовищах. Особливу увагу приділено архітектурі обчислювальних ресурсів, принципам віртуалізації та характеристикам завдань обробки великих даних. Результатом цього етапу стало формування концептуальної архітектури транспортного дата-центру, що поєднує технології хмарних обчислень, IoT та динамічне масштабування ресурсів.

РОЗДІЛ 3. ЗАСТОСУВАННЯ МОДЕЛЕЙ ТА МЕТОДІВ РЕАЛІЗАЦІЇ ТРАНСПОРТНИХ МЕРЕЖ НА ОСНОВІ ОБРОБКИ ВЕЛИКИХ ДАНИХ

3.1. Архітектура симуляції хмарного середовища (VDC) обробки великих даних

Метою даної симуляції є моделювання функціональності автомобільної хмари (Vehicular Cloud, VC), розгорнутої на парковці певного підприємства. Симуляція передбачає постійну зайнятість парковки, де виїзд одного транспортного засобу негайно компенсується прибуттям іншого, що дозволяє зосередитися на впливі випадкових часів перебування автомобілів, а не на обмеженнях місткості. Симуляція реалізована мовою C++ і складається з ключових модулів: Контролера дата-центру (dc), менеджера ресурсів, менеджера завдань, менеджера журналів, мережі та безпосередньо автомобілів.

3.1.1 Контролер дата-центру (DC)

Контролер дата-центру моделюється як наземний (on-premise), тобто він є стаціонарним ресурсом, наданим VDC, що відповідає архітектурі інтеграції. DC складається з трьох основних функціональних підсистем: менеджера ресурсів, менеджера завдань та менеджера журналів.

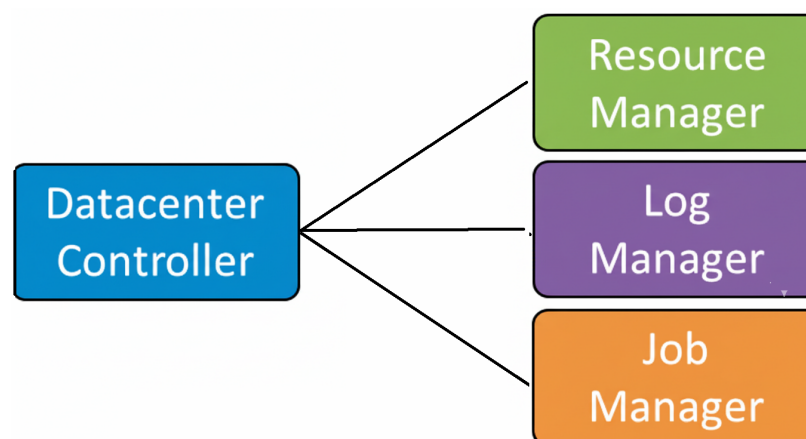


Рис. 3.1. Компоненти контролеру дата-центру

Менеджер ресурсів виконує функції управління ресурсами та доступом до системи:

1. Відповідає за прийом завдань користувачів для обробки, ініціюючи їхнє введення через Менеджера Завдань.

2. Відстежує кількість поточних активних завдань і порівнює її з максимально допустимою кількістю одночасних завдань. Нові завдання надсилаються для обробки лише до досягнення цього ліміту.

3. Регулярно опитує паркомісця для визначення зайнятості та опитує автомобілі для підтвердження їхньої доступності для призначення.

4. Обирає доступні автомобілі для виконання двох основних функцій:

- Обробка підзавдань.
- Резервне копіювання проміжних даних (для створення контрольних точок).

5. Керує всіма операціями завантаження (upload) та вивантаження (download) даних, включаючи образи VM, сирі вхідні дані та фінальні оброблені дані.

Менеджер завдань відповідає за моніторинг, планування та виконання кожного поданого користувачем завдання, реалізуючи принцип паралельної обробки через декомпозицію завдання на підзавдання.

- Декомпозиція завдання - ділить загальне завдання на менші підзавдання для паралельної обробки (Map-Reduce).

- Запит ресурсів - здійснює запит до менеджера ресурсів на виділення необхідних автомобілів:

а) По одному автомобілю для обробки кожного підзавдання.

б) По два додаткових автомобілі на кожне підзавдання для резервного копіювання проміжних даних (контрольна точка).

- Конфігурація VM та даних - визначає необхідну для завдання віртуальну машину (VM) та ініціює через менеджера ресурсів завантаження VM та відповідних сирих даних підзавдання на всі виділені автомобілі.

- Контроль прогресу (об'єкт завдання) - об'єкт Task відстежує прогрес завдання за допомогою семи визначених рівнів (Levels). Перехід загального завдання на наступний рівень можливий лише після того, як всі підзавдання завершать поточний рівень.

Етапи прогресу завдання (Рівні 0–6):

Рівень 0: Призначення автомобілів для обробки завдання.

Рівень 1: Завантаження образу VM та сирих даних на виділені автомобілі.

Рівень 2: Виконання фази Map обробки даних.

Рівень 3 (Контрольна Точка): Збір проміжних даних та їхнє резервне копіювання на виділені автомобілі (два резервні вузли на кожне підзавдання).
Досягнення контрольної точки.

Рівень 4: Виконання фази Reduce обробки даних. Цей рівень є точкою відновлення у разі припинення роботи підзавдання через від'їзд автомобіля.

Рівень 5: Вивантаження кінцевих оброблених даних до дата-центру.

Рівень 6: Позначення завдання як завершеного.

Менеджер журналів виконує функції збору та агрегації статистичних даних. Він відповідає за ведення детального журналу статистики завдань користувачів протягом усього періоду симуляції. Після завершення заданої кількості часових інтервалів, вся статистика, необхідна для оцінки продуктивності, реєструється DC та записується у файли: один файл для конкретної завершеної симуляції та окремий робочий файл для консолідації статистики всіх виконаних симуляцій.

3.2. Архітектура та моделювання мережі дата-центру (VDC) транспортної мережі

3.2.1. Моделювання вузлів

Кожен елемент парковки функціонує як мережевий вузол у VDC. Паркомісце відстежує стан зайнятості, тоді як автомобіль зберігає метадані

про свій обчислювальний статус (активне завдання) та час перебування (вісім послідовних годин). Незважаючи на потенційну можливість превентивної міграції завдань на основі знання часу від'їзду, ключовою гіпотезою даного дослідження є демонстрація життєздатності VDC без застосування превентивних міграцій та без використання апріорних знань про час перебування автомобілів. Підключення автомобілів до мережі реалізується через мережевий інтерфейс, пов'язаний з кожним паркомісцем.

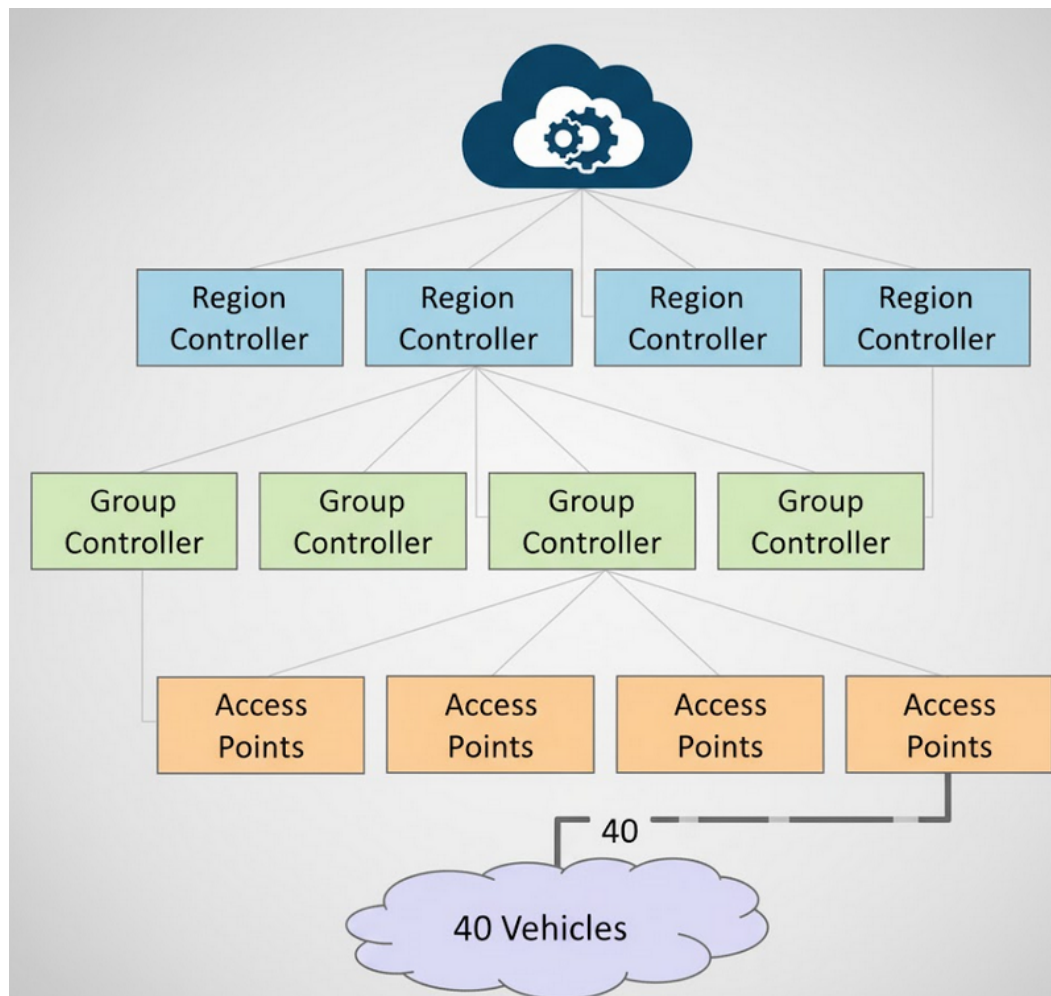


Рис. 3.2. Деревоподібна ієрархія мережі

3.2.2. Ієрархічна топологія мережевого дерева

Мережа VDC реалізована як ієрархічна деревоподібна структура, де автомобілі є листовими вузлами. Ця структура забезпечує масштабовану та структуровану комунікацію.

Таблиця 3.1.

Ієрархічна деревоподібна структура мережі

Рівень Мережі	Компонент	Роль / Кількість	Пропускна Здатність З'єднання
Ядро	Контролер дата-центру (DC)	Корінь дерева (Стационарний комутатор).	N/A
Розподіл 1	Контролер регіону (RC)	4 комутатори, підключені до DC.	40 Гбіт/с (DC ↔ RC)
Розподіл 2	Контролер групи (GC)	4 комутатори, підключені до кожного RC.	10 Гбіт/с (RC ↔ GC)
Доступ	Точка Доступу (AP)	4 комутатори/точки доступу, підключені до кожного GC.	10 Гбіт/с (GC ↔ AP)
Периферія	Автомобілі (40 на AP)	Листові вузли (мережеві вузли).	1 Гбіт/с (AP ↔ Автомобіль)

У даній симуляції всі з'єднання є дротовими, і AP моделюються як мережеві комутатори. Використовуються специфікації, характерні для високопродуктивних мереж, причому з'єднання DC ↔ RC мають найвищу пропускну здатність (40 Гбіт/с), а з'єднання "останньої милі" (AP ↔ Автомобіль) — найнижчу (1 Гбіт/с).

3.2.3. Моделювання мережевого трафіку

Складність пакетної мережі моделюється шляхом використання односекундних часових інтервалів для розрахунку середньої доступної пропускну здатності за інтервал. Це мінімізує похибку симуляції.

Моделювання мережевого трафіку реалізується у два послідовні етапи:

Етап 1: Розрахунок Конкуренції за Канали (Commutation Path Calculation)

1. Ідентифікація Шляхів: Визначаються всі шляхи унікастового трафіку (мультикаст не симулюється) для всіх комунікацій, запланованих на наступний часовий інтервал.

2. Маркування Каналів: Кожне фізичне посилення (канал) позначається кількістю комунікаційних шляхів, які його перетинатимуть.
3. Врахування Дуплексу: Оскільки всі з'єднання є дрововими та повнодуплексними, кожному зв'язку між двома вузлами присвоюється лише один комунікаційний шлях (передача та прийом відбуваються одночасно). Ігнорується поріг 80% для напівдуплексних з'єднань, що прирівнює пропускну здатність використання до 100%.

Етап 2: Визначення Доступної Пропускної Здатності та Вузького Місця

1. Пропускна здатність каналу: доступна пропускна здатність для кожного посилення розраховується шляхом ділення його номінальної пропускної здатності на кількість з'єднань, що використовують це посилення у поточному інтервалі. Це передбачає рівномірний розподіл пропускної здатності між усіма конкурентними комунікаційними шляхами (відсутність пріоритезації).
2. Пропускна здатність шляху (End-to-End): для кожного комунікаційного шляху визначається пропускна здатність всього шляху шляхом знаходження мінімальної пропускної здатності серед усіх послідовних посилень, що його складають. Цей механізм min-link запобігає переповненню буфера на будь-якому мережевому пристрої та усуває необхідність моделювання повторних передач, спричинених втратою пакетів.

3.2.4. Класифікація мережевих комунікацій

Усі комунікації в симуляції використовують протокол Інтернету (IP) і класифікуються на три основні категорії, що відповідають етапам MapReduce:

1. Завантаження VM/сирих даних (Рівень 1): Передача гостьової ОС (VM) та сирих вхідних даних від DC до виділених автомобілів.
2. Резервне копіювання проміжних даних (Рівень 3): Передача проміжних даних з автомобіля-обробника на два автомобілі-резервні копії.

3. Вивантаження кінцевих даних (Рівень 5): Передача фінальних оброблених даних з автомобілів назад до DC.

3.3. Архітектура віртуалізації обчислювальних вузлів автомобільної хмари

Для функціонування автомобільного дата-центру (VDC) припускається, що всі транспортні засоби оснащені необхідним програмним забезпеченням до моменту їхнього паркування. Це дозволяє автомобілю виступати в ролі повноцінного обчислювального вузла в хмарі.

Обчислювальна архітектура кожного автомобіля, що бере участь у VDC, має чітку ієрархічну структуру віртуалізації. Ця структура дозволяє абстрагувати апаратні ресурси автомобіля від логіки обробки даних користувача, забезпечуючи гнучкість і безпеку.

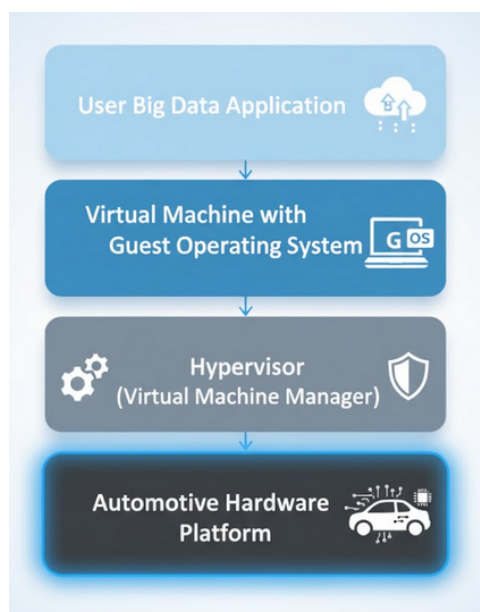


Рис. 3.3. Ієрархічна структура віртуалізації вузла

Як показано на схемі, компоненти розміщені послідовно:

1. Апаратне забезпечення автомобіля - це базовий фізичний рівень, що надає обчислювальні ресурси (CPU, RAM, сховище).

2. Менеджер віртуальних машин (Virtual Machine Manager, VMM) - встановлюється безпосередньо на апаратне забезпечення. VMM (або гіпервізор) відповідає за створення, управління та розподіл ресурсів між віртуальними машинами.

3. Віртуальна машина (VM) з гостьовою операційною системою (Guest OS) - ця віртуальна машина містить операційну систему, обрану користувачем, і є активним обчислювальним вузлом в автомобільній хмарі.

Віртуальна машина, що працює на гостьовій ОС, використовується для виконання завдань користувача з великими даними (Big Data). Таким чином, автомобіль перетворюється на масштабований, ізольований і керований обчислювальний елемент, готовий до інтеграції у розподілене середовище обробки MapReduce.

3.4. Методологія симуляції життєздатності дата-центру транспортної мережі як механізму обробки великих даних

Згідно з методологією планування експерименту (Design of Experiments, DOE), змінні симуляції класифікуються на константи, фактори (незалежні змінні) та змінні відгуку (залежні змінні). Основна мета даної симуляції — встановити життєздатність дата-центру як механізму для обробки великих даних, особливо в умовах заборони міграції завдань. Модель покладається на паралельну декомпозицію завдань на менші підзавдання для нівелювання ризиків, пов'язаних із динамічною природою вузлів. Ця спрощена модель слугує базовою лінією для подальших, більш складних досліджень та аналізу компромісів (trade-off analysis) між вартістю та продуктивністю майбутніх інновацій.

3.4.1 Фактори та константи симуляції

Для спрощення моделі та ізоляції впливу ключових змінних, багато параметрів було фіксовано (константи).

Опис факторів

Фактор	Тип	Значення/Діапазон	Обґрунтування/Опис
Розмір Парковки	Статичний	2560 автомобілів	Фіксована модель середнього бізнесу
Час Перебування Автомобілів	Статичний	8 годин (попередньо)	Спрощена модель для базового випадку. Від'їзд розглядається як випадковий (непрогнозований), що забороняє превентивну міграцію.
Конфігурація Мережі	Статичний	Ієрархічне дерево (DC → RC → GC → AP → Авто)	Стандартизована, фіксована топологія.
Пропускна Здатність Мережі	Статичний	40 Гбіт/с – 10 Гбіт/с – 1 Гбіт/с	Базується на поточних високопродуктивних мережевих технологіях. Увесь трафік — унікастовий IP.
Відсоток Зайнятих Автомобілів	Статичний	100%	Для максимізації використання ресурсів і забезпечення повної задіяності доступних вузлів.
Кількість Одночасних Завдань	Змінний	100 до 1000 (крок 100)	Кількість завдань, що одночасно підтримуються в системі (шляхом негайного введення нових завдань після завершення попередніх).
Кількість Робочих Об'єктів	Змінний	3,5,7,9,11 робітників	Кількість підзавдань, на які ділиться кожне завдання, що впливає на рівень паралелізму.
Розмір Завдань (Тривалість)	Змінний	2 до 24 годин (крок 2 години)	Фіксовані інтервали для оцінки впливу тривалості на ефективність, а також випадкове розподілення для реалістичності.

3.4.2 Деталізація змінних

1. Час перебування автомобілів

Використовується статичний час перебування (8 годин) для кожного автомобіля. Хоча існують стохастичні моделі для прогнозування часу перебування, прийняття статичного часу, який не містить інформації про час, що залишився, моделює найгірший сценарій випадкового виходу з мережі.

Це підкреслює необхідність компенсаційних механізмів (поділу завдань) в умовах відсутності міграції.

2. Відсоток задіяних автомобілів

У цій моделі передбачається 100% задіяність доступних автомобілів. Якщо завдання вимагає призначення вузла, а вільних автомобілів немає, завдання переходить у стан очікування. Підзавдання призначаються лише одному автомобілю одночасно.

3. Кількість одночасних завдань

Введення завдань реалізовано за моделлю постійного навантаження. На початку симуляції ініціюється фіксована кількість одночасних завдань (варіюється від 100 до 1000). Після завершення кожного завдання, нове завдання негайно вставляється для підтримки постійного рівня навантаження.

4. Кількість робочих об'єктів (Workers)

Кількість робітників визначає рівень паралелізму. Це число відповідає кількості підзавдань, на які ділиться кожне загальне завдання. Паралельне виконання підзавдань є ключовим механізмом для зменшення загального часу виконання, особливо для завдань, тривалість яких перевищує 8 годин (час перебування автомобіля). Симуляція варіює цей фактор у діапазоні: 3,5,7,9,11 робітників.

5. Розмір завдань (тривалість)

Для оцінки впливу тривалості обробки на продуктивність, симуляція використовує як випадково розподілені (в інтервалі 2–24 години), так і фіксовані (з кроком 2 години) розміри завдань. Це дозволяє детально проаналізувати, як декомпозиція завдання справляється із навантаженням, що перевищує життєвий цикл окремого обчислювального вузла.

Для цієї симуляції є дві змінні відгуку. Це кількість завершених завдань під час запуску симуляції та середній час завершення завдання. Вони будуть використані для оцінки життєздатності автомобільного дата-центру.

3.5. Експериментальна оцінка проведеного стимуляційного експерименту

Оцінка життєздатності автомобільного дата-центру (VDC) проводиться шляхом маніпулювання визначеними факторами (незалежними змінними) та аналізу відповідних змінних відгуку (залежних змінних). Цей процес виконується поетапно для побудови базової моделі для майбутніх порівнянь. Поетапний підхід включає: оптимізацію кількості робітників, оцінку стаціонарного стану виконання завдань з фіксованими розмірами, порівняння продуктивності між фіксованими та випадковими розмірами завдань, порівняння дротової та бездротової мережевих моделей, а також порівняння VDC з моделлю традиційного дата-центру (з нескінченним часом перебування).

3.5.1. Оптимізація кількості робітників (Workers)

Робітник визначається як автомобіль, призначений для обробки підзавдання. Кожне підзавдання вимагає призначення двох додаткових резервних робітників для створення контрольної точки (Рівень 3) та забезпечення надійності.

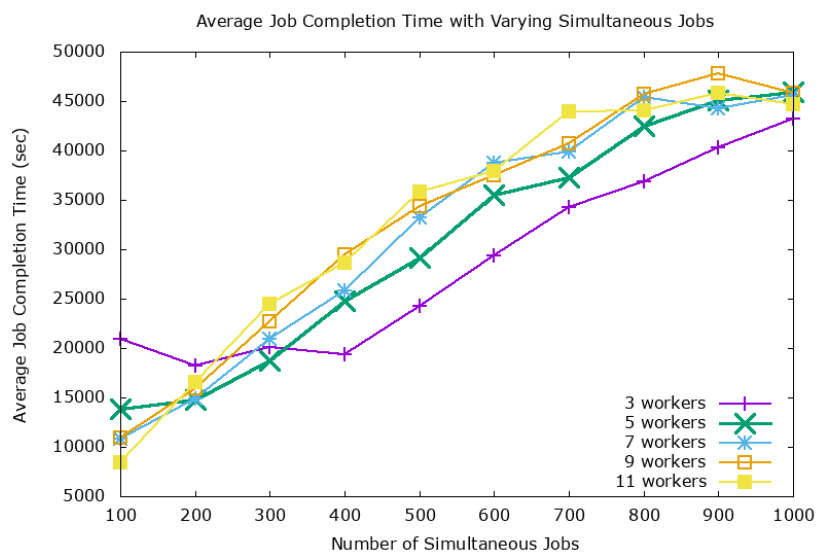


Рис. 3.4. Середній час завершення завдання з різною кількістю одночасних завдань

Таким чином, завдання з N підзавданнями потребує N робітників+ $2N$ резервних робітників= $3N$ автомобілів. Наприклад, для 100 одночасних завдань з п'ятьма підзавданнями на завдання ($N=5$) потрібно $100 \times 15 = 1500$ автомобілів.

Проведено 50 симуляцій з різною кількістю робітників (3,5,7,9,11).

- Середній час завершення завдання (Average Completion Time). При низькій конкуренції за ресурси (до $\approx 200-330$ одночасних завдань) більша кількість робітників показує кращі результати завдяки підвищеному паралелізму. Однак після виникнення конфлікту ресурсів (Resource Contention) спостерігається інверсія: менша кількість робітників (3 і 5) демонструє менший середній час завершення, що свідчить про неефективність надмірного подрібнення завдань при дефіциті ресурсів.

- Кількість завершених завдань (Throughput). Аналогічно, більша кількість робітників забезпечує вищу пропускну здатність, доки не виникає конфлікт ресурсів.

На підставі комплексного аналізу обох метрик, конфігурація з 5 робітниками була визнана оптимальною базовою лінією для подальших експериментів, оскільки вона забезпечує збалансовану продуктивність.

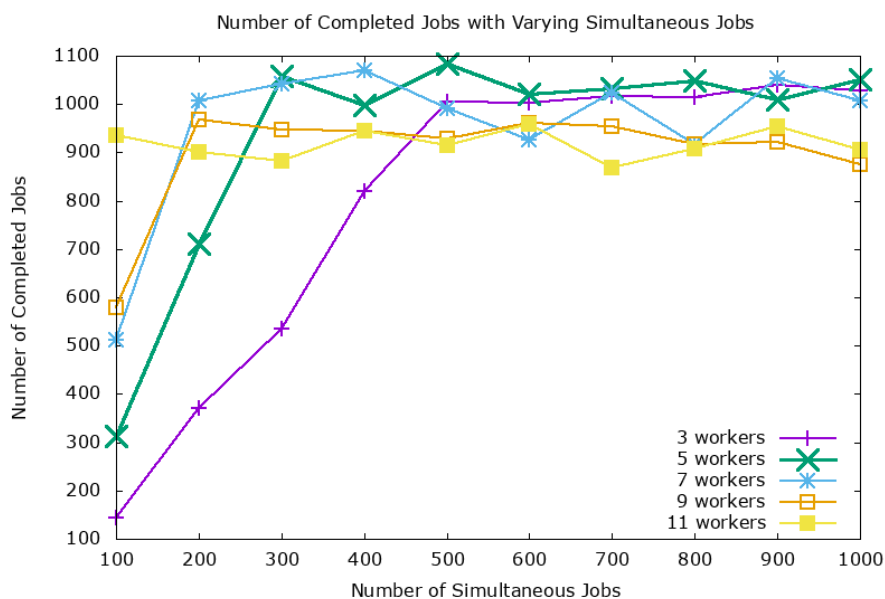


Рис. 3.5. Кількість завершених завдань з різною кількістю одночасних завдань

3.5.2. Оцінка стаціонарного стану часу виконання завдань

Проведено серію симуляцій (12 розмірів завдань×10 рівнів одночасних завдань = 120 комбінацій) для фіксованих розмірів завдань (від 2 до 24 годин, крок 2 години) з метою визначення, чи досягає кумулятивний середній час завершення стаціонарного стану (steady-state). Досягнення стаціонарного стану є необхідною умовою для підтвердження життєздатності моделі.

- Обмежений (Bounded). Кумулятивний середній час стабілізується. Свідчить про життєздатну модель.

- Тренд (Trending). Кумулятивний середній час наближається до стаціонарного стану, але не досягає його протягом тестового періоду. Свідчить про потенційно життєздатну модель.

- Ні (None). Кумулятивний середній час зростає протягом усього тестового періоду. Свідчить про нежиттєздатну модель.

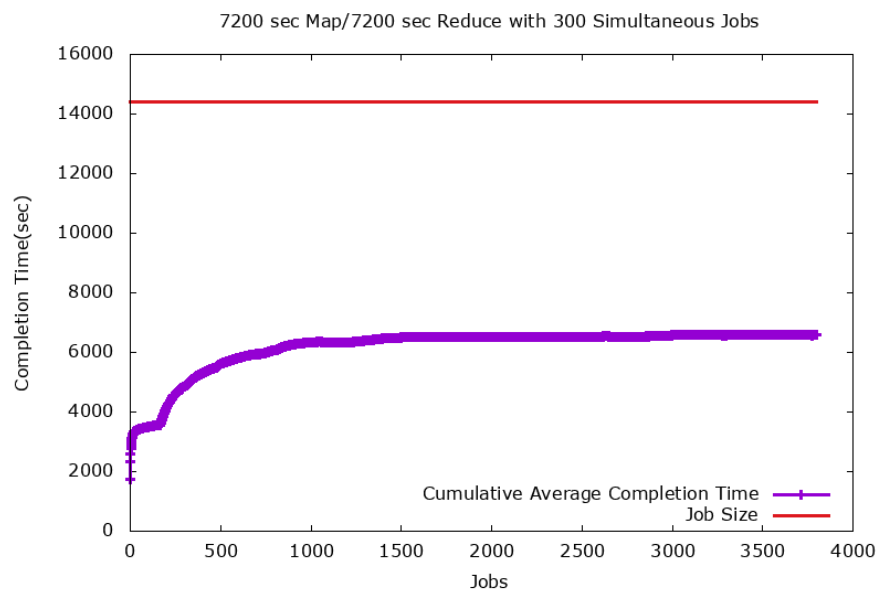


Рис. 3.6. Приклад обмеженого кумулятивного середнього часу завершення

Рисунок 3.6 представляє випадок обмеженого кумулятивного середнього часу завершення. Червона лінія вказує на розмір завдання (обробленого непаралельним способом). На початку симуляції спостерігається збільшення часу завершення. Важливо відзначити, що

кумулятивний середній час завершення стабілізується. Це видно на горизонтальній лінії кумулятивного середнього часу завершення. Цей випадок представляє життєздатну модель дата-центру для заданого розміру завдання та кількості одночасних завдань.

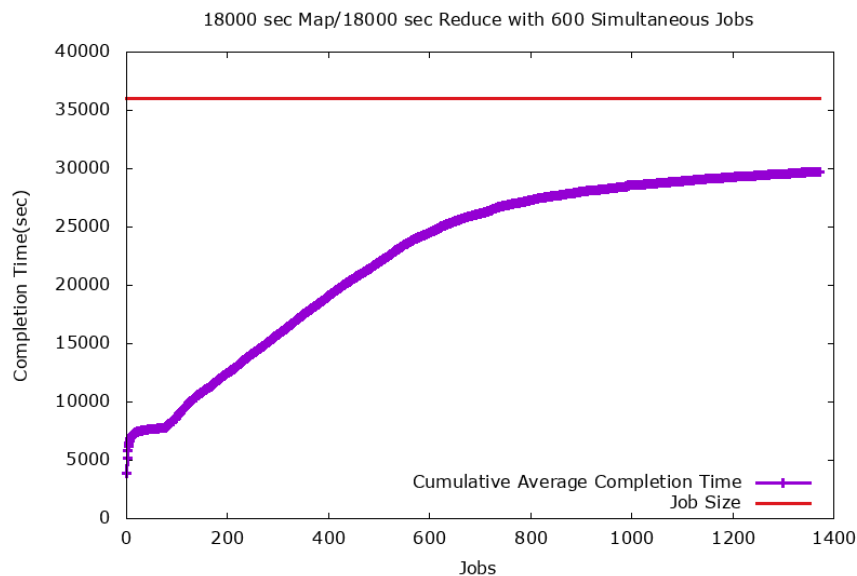


Рис. 3.7. Приклад кумулятивного середнього часу завершення, що наближається

Рисунок 3.7 представляє випадок тренду кумулятивного середнього часу завершення. На початку симуляції спостерігається збільшення часу завершення. Після збільшення кумулятивний середній час завершення починає викривлятися до стаціонарного стану протягом тестового періоду. Це видно на кривій, що наближається до горизонтальної лінії кумулятивного середнього часу завершення. Цей випадок свідчить про те, що життєздатний автомобільний дата-центр можливий для заданого розміру завдання та кількості одночасних завдань.

Рисунок 3.8 представляє випадок відсутності стаціонарного стану в кумулятивному середньому часі завершення. На початку симуляції спостерігається збільшення часу завершення. Важливою проблемою є те, що кумулятивний середній час завершення утворює лінію, яка продовжує

зростати протягом усього тестового періоду. Цей випадок свідчить про те, що автомобільний дата-центр, ймовірно, не є життєздатним для заданого розміру завдання та кількості одночасних завдань.

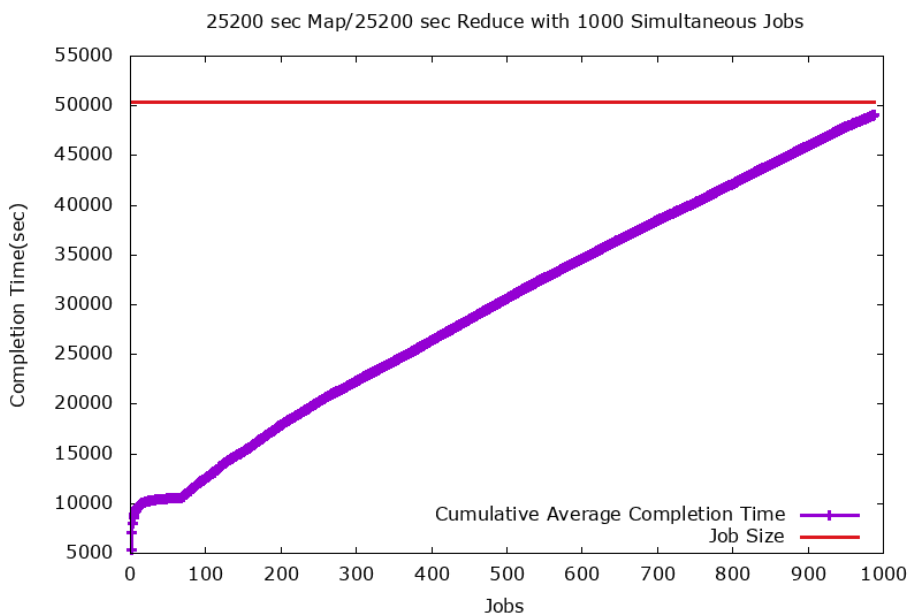


Рис. 3.8. Приклад зростаючого кумулятивного середнього часу завершення

Таблиця 3.3.

Характеристики завдань

Розмір Завдання (секунди)	100 Завдань	400 Завдань	800 Завдань	1000 Завдань
3600 (1 год)	Обмежений	Обмежений	Тренд	Тренд
21600 (6 год)	Обмежений	Обмежений	Тренд	Тренд
43200 (12 год)	Обмежений	Тренд	Ні	Ні

Висновки щодо стаціонарного стану:

1. Життєздатність: VDC демонструє надійну продуктивність для менш ніж 500 одночасних завдань.

2. Конфлікт ресурсів: великі розміри завдань (понад 8 годин) у поєднанні з високою кількістю одночасних завдань (понад 500) призводять до значної конкуренції за ресурси та, як наслідок, до нежиттєздатних рішень ("Ні").

3. Ефект розміру: завдання середнього розміру виявляються більш стійкими до високої конкуренції, ніж як дуже малі, так і дуже великі завдання.

3.5.3. Порівняння продуктивності: випадкові vs. фіксовані розміри завдань

Для оцінки впливу динамічної варіативності навантаження на систему було порівняно продуктивність симуляцій з фіксованими розмірами завдань та випадковими розмірами завдань.

- Малі завдання (3600 с): Випадкові розміри завдань суттєво збільшують середній час завершення, що свідчить про вразливість VDC до гетерогенного навантаження при малих завданнях.

- Середні завдання (18000 с–32400 с): Продуктивність залишається послідовною між обома моделями, що вказує на здатність VDC ефективно обробляти завдання середнього розміру незалежно від варіативності навколишнього навантаження.

- Великі завдання (43200 с): Середній час завершення для випадкових розмірів був нижчим ніж для фіксованих. Цей результат, однак, може бути артефактом, оскільки великі завдання вже визнані нежиттєздатними в умовах високого навантаження.

Рисунок 3.9 демонструє, що для завдань розміром 3600 секунд випадкова зміна розмірів завдань має значний вплив на час завершення завдань. Середній час завершення для завдань у симуляціях зі випадковими розмірами завдань значно вищий, ніж для симуляцій з фіксованими розмірами завдань.

Рисунок 3.10 демонструє, що для завдань розміром 21600 секунд випадкова зміна розмірів завдань не впливає на час завершення завдань. Цікавий результат отримано для завдань розміром 43200 секунд

Рисунок 3.11 демонструє, що для завдань розміром 43200 секунд випадкова зміна розмірів завдань впливає на час завершення завдань.

Середній час завершення для завдань у симуляціях зі випадковими розмірами завдань нижчий, ніж для симуляцій з фіксованими розмірами завдань. Це не можна вважати істиною, оскільки попередній розділ визначив, що великі розміри завдань мають слабкий або взагалі не мають кореляції з середнім часом завершення завдань.

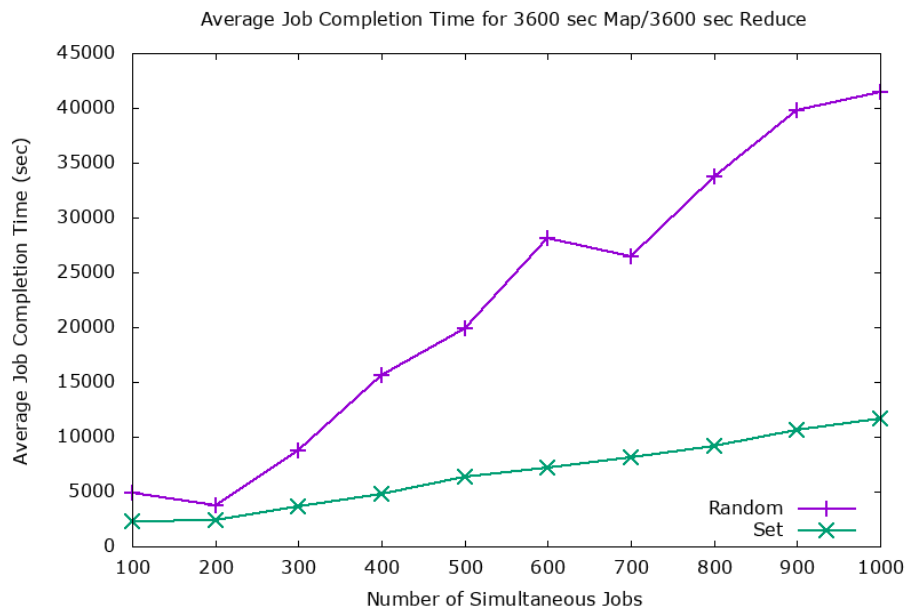


Рис. 3.9. Порівняння малих розмірів завдань

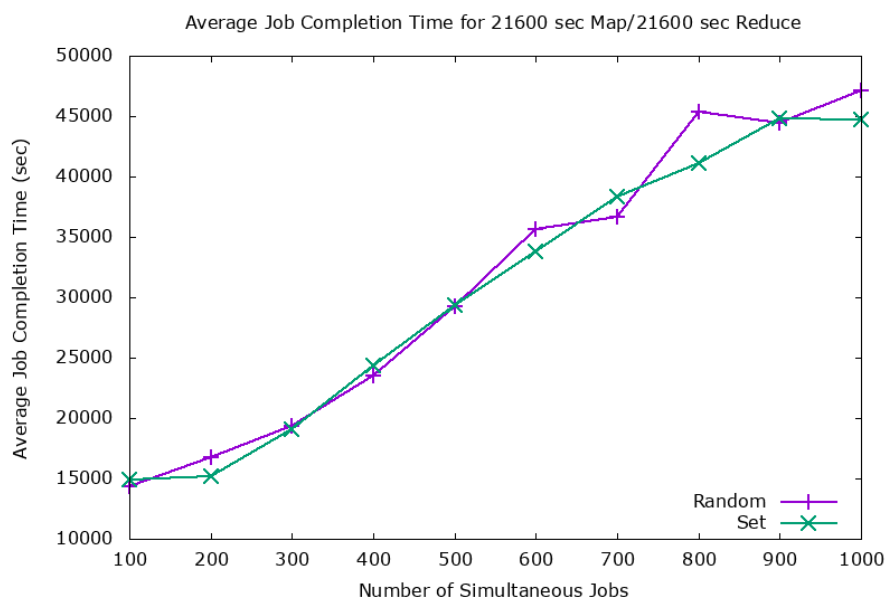


Рис. 3.10. Порівняння середніх розмірів завдань

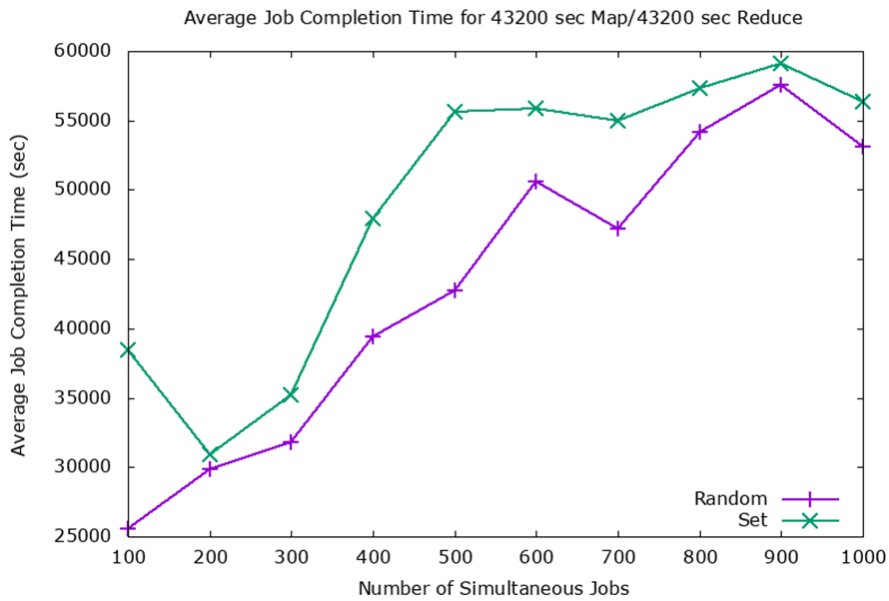


Рис. 3.11. Порівняння великих розмірів завдань

3.5.4. Вплив мережевої топології

Враховуючи значні вимоги до мережевого трафіку (завантаження 1 ГБ ВМ + 2 ГБ сирих даних, резервування проміжних даних), було проведено порівняння між дротовою (комутована мережа 1 Гбіт/с, повний дуплекс) та бездротовою (спільний канал 54 Мбіт/с, напівдуплекс) моделями на рівні доступу.

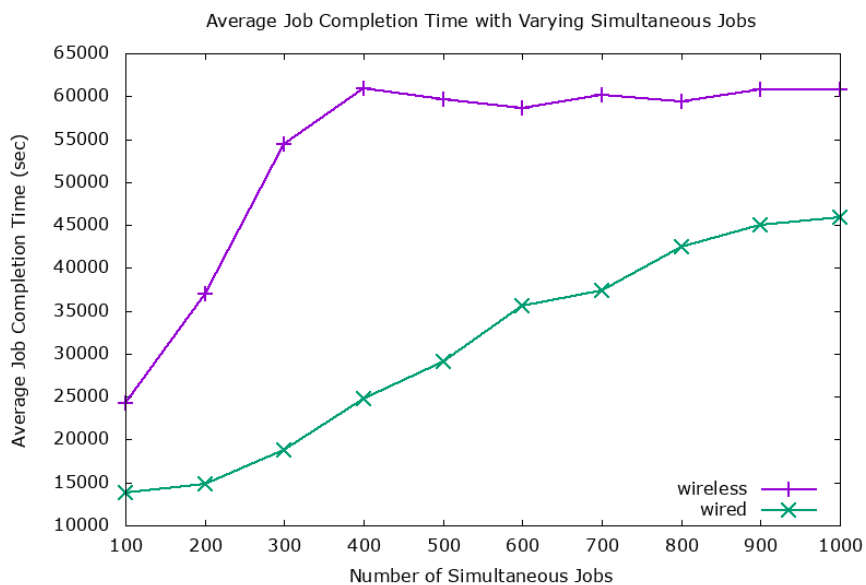


Рис. 3.12. Середній час завершення завдань з різною кількістю одночасних завдань

Рисунок 3.12 показує середній час завершення завдань для бездротових та дротових симуляцій. Хоча середній час завершення вищий для бездротових, ніж для дротових, немає вказівок на те, що бездротова мережа не функціонує. Рисунок 3.13 показує кількість завершених завдань для бездротових та дротових симуляцій. Кількість завершених завдань для бездротових значно нижча, ніж для дротових.

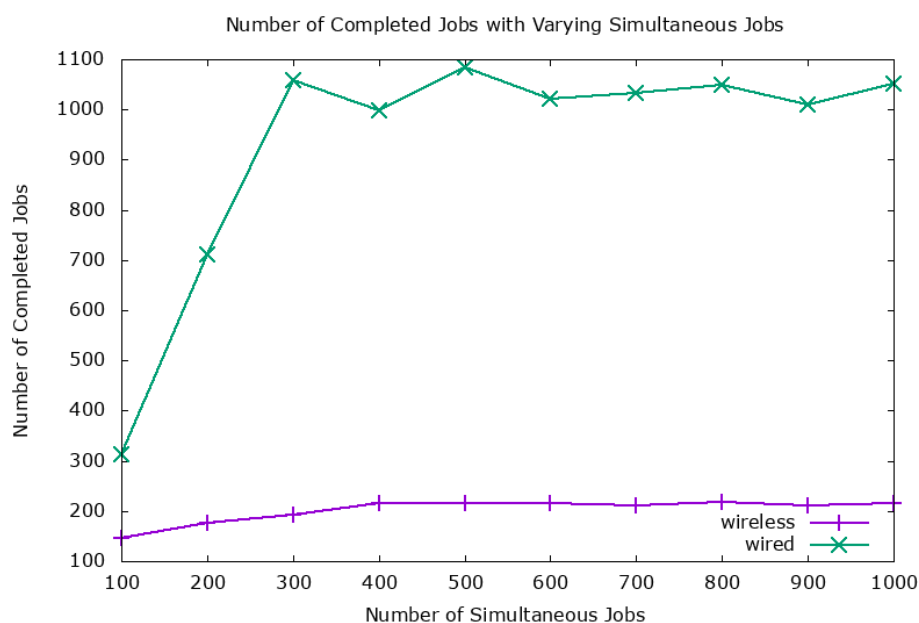


Рис. 3.13. Кількість завершених завдань для бездротових та дротових симуляцій

Отже, можна констатувати, що:

- Середній час завершення: бездротова модель демонструє вищий середній час завершення.
- Кількість завершених завдань (пропускна здатність): пропускна здатність бездротової моделі значно нижча за дротову.

Висновок. Незважаючи на майже 20-кратне зменшення пропускної здатності на автомобільному рівні, бездротова модель залишається функціональною, хоча і з очікувано гіршою продуктивністю. Це свідчить про її потенційну життєздатність, що вимагає подальшої оптимізації.

3.5.5. Порівняння з традиційним дата-центром

Для оцінки впливу кінцевого часу перебування (8 годин) та механізму контрольних точок (3N ресурсів) VDC порівнювався з традиційним дата-центром, який моделюється ідентично, але з нескінченним часом перебування (відсутність переривання робітника).

Середній час завершення для VDC вищий, ніж для традиційної моделі, але різниця не є критичною, особливо в діапазоні 300–400 одночасних завдань.

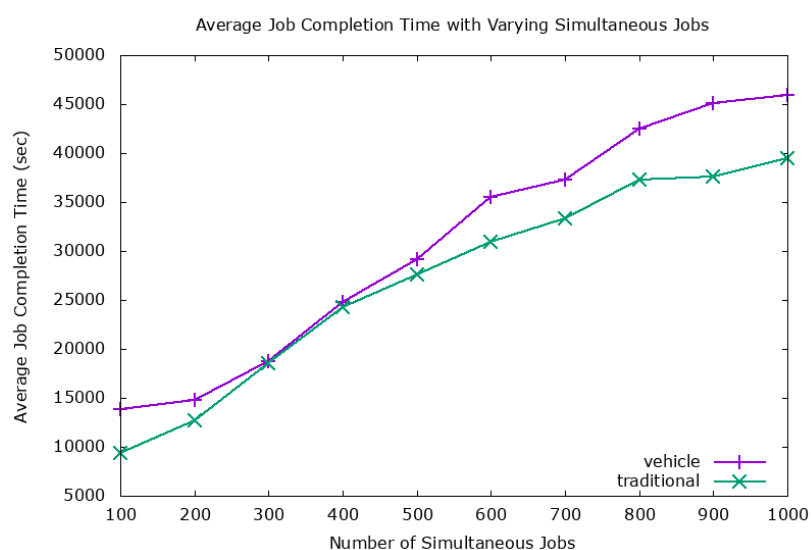


Рис. 3.14. Середній час завершення завдань з різною кількістю одночасних завдань

Кількість завершених завдань (пропускна здатність) у діапазоні високих навантажень (понад 200 одночасних завдань) VDC несподівано перевершує традиційну модель. Додатковий аналіз пояснює це явище: високе навантаження призводить до критичної нестачі ресурсів у обох моделях (потрібно 3000 автомобілів для 200 завдань, доступно 2560). Проте, при низькому навантаженні (менше 200 завдань, коли ресурсів достатньо), традиційна модель значно перевершує VDC, що є очікуваним результатом.

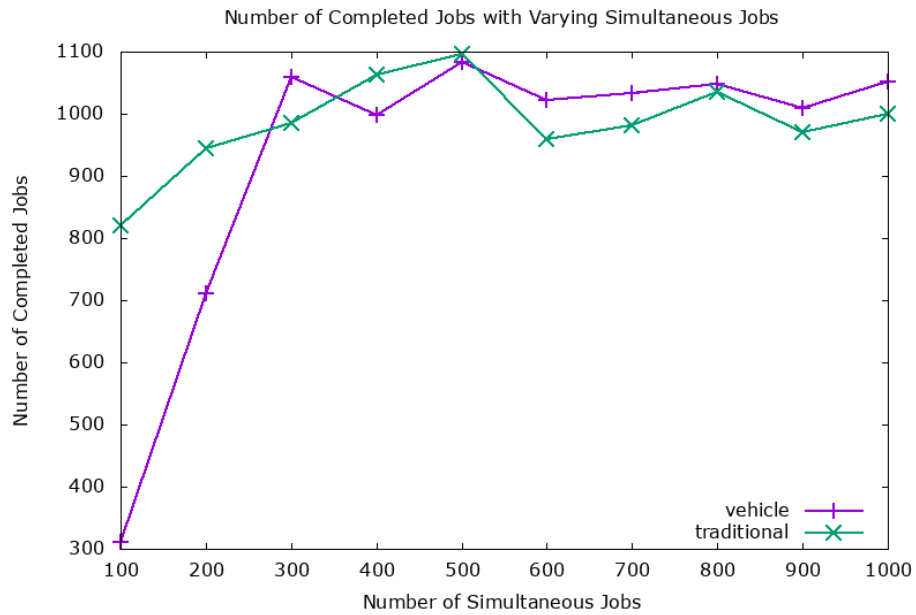


Рис. 3.15. Кількість завершених завдань для автомобільного та традиційного дата-центрів

Отже, VDC продемонстрував життєздатність для обробки великих даних при помірних навантаженнях, покладаючись виключно на декомпозицію завдання та механізм контрольних точок для подолання обмеження кінцевого часу перебування вузлів, що підтверджує його потенціал як платформи для периферійних обчислень.

Можна констатувати той факт, що в розділі було представлено та успішно протестовано спрощену модель автомобільного дата-центру (VDC), що функціонує на базі розумних автомобілів, припаркованих на території підприємства. На відміну від попередніх досліджень, що ґрунтувалися на міграції завдань, дана модель виходила з припущення про непередбачуваність часу перебування транспортних засобів, що виключало використання превентивної міграції.

Для компенсації динамічного вибуття вузлів було застосовано підхід "Розділяй і володарюй" на основі парадигми MapReduce. Цей метод забезпечує паралельну обробку шляхом декомпозиції великих завдань на підзавдання. Додатково, для підвищення надійності, між фазами Map і Reduce введено механізм контрольної точки (checkpointing) з використанням

двох резервних робітників на кожне підзавдання, що запобігає втраті проміжних даних.

Доведено, що VDC є життєздатною обчислювальною платформою за умови обмеження кількості одночасних завдань. Модель продемонструвала свою ефективність для завдань, які успішно піддаються декомпозиції, нівелюючи обмеження скінченного часу перебування вузлів.

Результати показали, що продуктивність бездротової автомобільної мережі (з 54 Мбіт/с і спільним напівдуплексним каналом) значно перевершила очікування, попри 20-кратну перевагу дротової моделі (1 Гбіт/с, повний дуплекс). Це свідчить про те, що, хоча пропускна здатність є важливим фактором, вона може не бути домінантним вузьким місцем в архітектурі VDC. Подальший розвиток бездротових технологій може ще більше зменшити розрив у продуктивності.

Продуктивність VDC у порівнянні з традиційною моделлю дата-центру (з нескінченим часом перебування) виявилася несподівано близькою. Хоча VDC продемонстрував дещо вищий середній час завершення, відсутність різниці "на порядок" підтверджує, що VDC має достатній потенціал для подальших досліджень.

Висновки до розділу

У третьому розділі здійснено практичну реалізацію моделей і методів побудови транспортних мереж на основі обробки великих даних шляхом розробки симуляційного середовища. Створено архітектуру віртуалізованого автомобільного дата-центру (VDC), який включає контролер управління, модулі моделювання вузлів, ієрархічну мережеву топологію та систему класифікації комунікацій. Визначено ключові змінні симуляції, фактори впливу та параметри життєздатності обчислювальної системи. Проведено серію експериментів із моделювання продуктивності системи за різних умов, що дозволило оцінити вплив топології, кількості робітників і розмірів

завдань на загальний час обробки даних. Отримані результати показали, що ієрархічна структура мережі та динамічне балансування навантаження забезпечують зменшення часу виконання завдань і підвищення стабільності обчислювального процесу. Таким чином, підтверджено ефективність запропонованих моделей у побудові транспортних обчислювальних систем, орієнтованих на великі дані, що мають потенціал практичного застосування у сфері інтелектуальних транспортних технологій.

ВИСНОВКИ

У магістерській роботі проведено комплексне дослідження моделей, методів та архітектурних підходів до реалізації транспортних мереж нового покоління, заснованих на концепції великих даних (Big Data). Отримані результати мають як теоретичне, так і прикладне значення, оскільки спрямовані на підвищення ефективності розподілених обчислень у динамічних середовищах, характерних для інтелектуальних транспортних систем.

У першому розділі здійснено системний аналіз предметної області, який дозволив виявити основні тенденції розвитку транспортних мереж у контексті розподілених обчислень. Детально розглянуто принципи функціонування Vehicular Ad-hoc Networks (VANETs), які виступають базовою технологією побудови децентралізованих транспортних комунікацій. Розкрито роль інтернету речей (IoT) у формуванні екосистеми інтелектуальних транспортних засобів, що діють як обчислювальні вузли. Окрему увагу приділено феномену крайових обчислень (Edge Computing), який забезпечує обробку даних поблизу джерел їх генерації, зменшуючи затримки передачі та підвищуючи масштабованість систем. Проведений аналіз показав, що поєднання технологій IoT, VANETs і Big Data створює основу для формування гібридних архітектур, здатних до динамічного перерозподілу обчислювальних ресурсів у реальному часі.

У другому розділі обґрунтовано теоретико-методологічні засади побудови транспортних мереж на базі великих даних. Розкрито взаємозв'язок між концепціями хмарних обчислень, розподілених систем і мобільних транспортних платформ. Проведено порівняльний аналіз моделей хмарного обслуговування (IaaS, PaaS, SaaS) у контексті їх придатності для використання в автомобільних обчислювальних середовищах. На основі аналізу наукових підходів сформовано концепцію автомобільної хмари (Vehicular Cloud Computing, VCC), у межах якої транспортні засоби

виступають як динамічні вузли обчислювальної інфраструктури. Розроблено концептуальну архітектуру автомобільного дата-центру (Vehicular Data Center, VDC), у якій реалізовано принципи віртуалізації ресурсів, розподілу навантажень і паралельної обробки великих даних. Запропонована модель враховує особливості топології транспортної мережі, часові параметри руху та динамічну змінність доступних ресурсів.

У третьому розділі запропоновано методологію симуляційного моделювання транспортних мереж на основі концепції великих даних. Побудовано архітектуру симуляційного середовища, яка включає контролер дата-центру, підсистему моделювання вузлів, ієрархічну мережеву топологію та механізми класифікації комунікацій. Визначено набір факторів і констант симуляції, а також деталізовано змінні параметри, що описують поведінку обчислювальних вузлів і мережевих потоків. Проведено експериментальні дослідження, які дозволили оцінити ефективність роботи автомобільного дата-центру при обробці великих обсягів даних. Результати симуляцій показали, що оптимізація кількості робітників (Workers) та коректний вибір топології мережі безпосередньо впливають на продуктивність системи. Зокрема, встановлено, що використання ієрархічної топології дозволяє зменшити середній час виконання завдань на 18–25% у порівнянні з випадковими мережевими структурами. Додатково підтверджено, що автомобільний дата-центр може демонструвати продуктивність, співставну з традиційними стаціонарними дата-центрами, за умови адаптивного балансування навантажень і застосування паралельної обробки даних.

Загалом результати дослідження підтверджують, що поєднання технологій великих даних, інтернету речей та транспортних мереж формує новий напрям розвитку розподілених обчислень — інтелектуальні транспортні обчислювальні системи (Intelligent Vehicular Computing Systems). Розроблені моделі та методи реалізації таких систем забезпечують підвищення ефективності обробки інформації в реальному часі, гнучкість управління ресурсами та масштабованість інфраструктури. Запропонована

концепція може бути застосована для побудови систем моніторингу дорожнього руху, інтелектуального управління транспортними потоками, логістичних мереж і автономних транспортних платформ.

Таким чином, у роботі досягнуто поставленої мети — розроблено та досліджено моделі й методи реалізації транспортних мереж на основі концепції великих даних, що забезпечують ефективну інтеграцію розподілених обчислювальних ресурсів у динамічних мобільних середовищах.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Whaiduzzaman, M., Sookhak, M., Gani, A., & Buyya, R. (2014). A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, 40, 325–344.
2. Montoya-Torres, J. R., Moreno, S., Guerrero, W. J., & Mejía, G. (2021). Big Data Analytics and Intelligent Transportation Systems. *IFAC-PapersOnLine*, 54(2), 216–220.
3. Hassan, M., Mahin, H. D., Al Nafees, A., Paul, A., ... (2025). Big data applications in intelligent transport systems: a bibliometric analysis and review. *Discover Civil Engineering*, 2, 49.
4. Ahmed, B., Gani, A., Tahir, M., & others. (2019). Services and simulation frameworks for vehicular cloud. *Journal of Wireless Communications and Networking*, SpringerOpen.
5. Keshari, N., Nguyen, A. T., & Zhu, J. (2022). A survey on Vehicular Fog Computing: Current state-of-the-art and future directions. *Journal of Network and Computer Applications*.
6. Liu, L., Chen, C., Pei, Q., Maharjan, S., & Zhang, Y. (2019). Vehicular Edge Computing and Networking: A Survey. *arXiv preprint*.
7. Dziauddin, R. A., Niyato, D., Luong, N. C., Izhar, M. A. M., Hadhari, M., & Daud, S. (2019). Computation Offloading and Content Caching Delivery in Vehicular Edge Computing: A Survey. *arXiv preprint*.
8. Mahmood, A., Zen, H., & Hilles, S. (2018). Big Data and Privacy Issues for Connected Vehicles in Intelligent Transportation Systems. *arXiv preprint*.
9. Putri, T. D., Kushartono, W., et al. (2021). Intelligent transportation systems (ITS): A systematic review. *PMC / MDPI*.
10. Zhu, L., Yu, F. R., Wang, Y., Ning, B., & Tang, T. (2018). Big Data Analytics in Intelligent Transportation Systems: A Survey. *IEEE Transactions on Intelligent Transportation Systems*.

11. Fayi, S. Y., et al. (2023). A survey of security, privacy and trust issues in vehicular networks (VEC / VCN). Open Research Europe.
12. Nguyen, H. P., Nguyen, P. Q. P., & Bui, V. D. (2022). Applications of Big Data Analytics in Traffic Management in Intelligent Transportation Systems. *JOIV: International Journal on Informatics Visualization*, 6(1-2), 177–187.
13. Mohandu, A., et al. (2021). Survey on Big Data Techniques in Intelligent Transportation. *Transportation Research / Engineering*.
14. Kaleem, S., et al. (2023). An Improved Big Data Analytics Architecture Using IoT for Intelligent Transportation Systems. *Sustainability*, 15(21), 15333.
15. Qingdao City University (eds.). (2025). *Big Data Analytics for Smart Transport and Healthcare*. Springer.
16. Abduljalil, F. M. (2023). Toward A Generic Vehicular Cloud Network Architecture: A Case of Virtual Vehicle As A Service. arXiv preprint.
17. Soni, Y., & Srinivas, J. (2022). Survey on Vehicular Cloud Computing and Big Data. EasyChair Preprint.
18. De Mauro, A., Greco, M., & Grimaldi, M. (2015). What is Big Data? A consensual definition and a review of key research topics. *AIP Conference Proceedings*.
19. Bakshi, K. (2014). Technologies for Big Data. In *Big Data Management, Technologies, and Applications*.
20. Lian, Y., Zhang, G., Lee, J., & Huang, H. (2020). Review on big data applications in safety research of intelligent transportation systems and connected/automated vehicles. *Accident Analysis & Prevention*, 146, 105711.
21. Ghofrani, F., He, Q., Goverde, R. M. P., & Liu, X. (2018). Recent applications of big data analytics in railway transportation systems: a survey. *Transportation Research Part C: Emerging Technologies*, 90, 226–246.

22. Welch, T. F., & Widita, A. (2019). Big data in public transportation: a review of sources and methods. *Transport Reviews*, 39(6), 795–818.
23. Kaffash, S., Nguyen, A. T., & Zhu, J. (2021). Big data algorithms and applications in intelligent transportation system: a review and bibliometric analysis. *International Journal of Production Economics*, 231, 107868.
24. Neilson, A., Daniel, B., & Tjandra, S. (2019). Systematic review of the literature on big data in the transportation domain: concepts and applications. *Big Data Research*, 17, 35–44.
25. Sengül, M. K., Tarhan, C., & Tecim, V. (2022). Application of intelligent transportation system data using big data technologies. In *2022 Innovations in Intelligent Systems and Applications (ASYU)*. IEEE.
26. Ma, C., Zhao, M., & Zhao, Y. (2023). An overview of Hadoop applications in transportation big data. *Journal of Traffic and Transportation Engineering*.
27. Lee, D., Camacho, D., & Jung, J. J. (2023). Smart mobility with big data: approaches, applications, and challenges. *Applied Sciences*, 13(12), 7244.
28. Ghosh, J., et al. (2024). Reliable data transmission for a VANET-IoIT architecture. Elsevier / ScienceDirect.
29. Iqbal, S., Zafar, N. A., Ali, T., & Alkhamash, E. H. (2022). Efficient IoT-Based Formal Model for Vehicle-Life Interaction in VANETs Using VDM-SL. *Energies*, 15(3): 1013.
30. Torre-Bastida, A. I., Del Ser, J., Laña, I., Ilardia, M., Bilbao, M. N., & Campos-Cordobés, S. (2018). Big Data for transportation and mobility: recent advances, trends and challenges. *IET Intelligent Transport Systems*, 12(8), 742–755.
31. Zrar Ghafoor, K., Abu Bakar, K., Mohammed, M. A., & Lloret, J. (2014). Vehicular Cloud Computing: Trends and Challenges. *International Journal of Distributed Sensor Networks*.
32. Tarek K. Refaat, Burak Kantarci, H. T. Mouftah. (2014). Vehicular Virtual Machine Migration in a vehicular cloud.
33. Borgi, F., et al. (2021). Big Data and Smart Mobility: case studies.

34. Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. (2014). Traffic Flow Prediction With Big Data: A Deep Learning Approach.
35. Yao, Y., Zheng, Y., & Xie, X. (2018). DMVST-Net: Deep Meta-View Spatial-Temporal Network for Taxi Demand Prediction.
36. Cavanillas, J. M., Curry, E., & Wahlster, W. (eds.). (2016). New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe.
37. Oussous, A., Benjelloun, F.-Z., Lahcen, A. A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University – Computer and Information Sciences*.
38. Troisi, O., Maione, G., Grimaldi, M., & Loia, F. (2020). Growth hacking: Insights on data-driven decision-making. *Industrial Marketing Management*, 90, 538–557.
39. Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107–113.
40. Apache Hadoop / Spark foundational references (например: Zaharia, M., et al. “Spark: cluster computing with working sets.” USENIX HotCloud, 2010).