



Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерної інженерії та управління \_\_\_\_\_

Кафедра \_\_\_\_\_ електронних обчислювальних машин \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

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

Тип програми \_\_\_\_\_ освітньо-наукова \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)

Освітня програма \_\_\_\_\_ Системне програмування \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту \_\_\_\_\_ Шеховцов Олександр Володимирович \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Метод побудови віртуального кластера на граничному шарі  
Інтернету речей \_\_\_\_\_

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

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

3. Вхідні дані до роботи \_\_\_\_\_ Сети даних myntra\_products\_catalog.csv, countries.csv, states.csv,  
операційна система – Windows 11. Сервер баз даних – SQL Server.

Середовища розробки – Microsoft SQL Server Management Studio,  
Talend Open Studio, Power BI Desktop.

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

1) аналітичний огляд застосування комп'ютерів з обмеженими обчислювальними;

2) обчислювальними можливостями у мережах Інтернету речей;

3) обґрунтувати вибір методів і алгоритмів побудови віртуального кластера;

4) побудувати віртуальний розподілений кластер;

5) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 15

---

---

---

---

---

---

---

---

---

---

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

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

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз літературних джерел-	02.04.24-08.04.24	
2	Вибір та обґрунтування методики дослідження	09.04.24-16.04.24	
3	Вибір інструментальних засобів	17.04.24-22.04.24	
4	Розробка моделей протоколів	23.04.24-06.05.24	
5	Проведення експериментів	07.05.24-23.05.24	
6	Оформлення матеріалів кваліфікаційної роботи	24.05.24-03.06.24	
7	Подання кваліфікаційної роботи керівникові та її попередній захист	04.06.24-07.06.24	
8	Подання кваліфікаційної роботи на рецензування	08.06.24-12.06.24	

Дата видачі завдання 01 квітня 2024 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

Зав. кафедри. Коваленко А.А.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 86 с., 31 рис., 5 табл., 5 дод., 17 джерел.

ІНТЕРНЕТ РЕЧЕЙ, ГРАНИЧНИЙ ШАР, ОДНОПЛАТНИЙ КОМП'ЮТЕР, ЧОТИРИШАРОВА АРХІТЕКТУРА, ВІРТУАЛЬНИЙ КЛАСТЕР.

Предметом дослідження є моделі та методи віртуалізації граничного шару одноплатних комп'ютерів Інтернету речей. Об'єктом дослідження є процес організації віртуального кластеру пристроїв граничного шару одноплатних комп'ютерів Інтернету речей.

Метою дослідження є підвищення ефективності використання одноплатних комп'ютерів з обмеженими можливостями, розташованих на граничному шарі Інтернету речей. Визначені характерні особливості застосування комп'ютерів з обмеженими обчислювальними можливостями у мережах Інтернету речей. Обґрунтований вибір методів і алгоритмів побудови віртуального кластеру. Сформована архітектура віртуального кластера комп'ютерів з обмеженими ресурсами. Удосконалений метод віртуалізації граничного шару одноплатних комп'ютерів Інтернету речей за рахунок побудови віртуального кластеру гетерогенних пристроїв з обмеженими обчислювальними ресурсами.

## ABSTRACT

Master's thesis: 86 pages, 29 figures, 11 tables, 3 appendices, 27 sources.

INTERNET OF THINGS, BOUNDARY LAYER, SINGLE BOARD COMPUTER, FOUR-LAYER ARCHITECTURE, VIRTUAL CLUSTER.

The subject of the research is the models and methods of virtualization of the boundary layer of single-board computers of the Internet of Things. The object of the study is the process of organizing a virtual cluster of boundary layer devices of single board computers of the Internet of Things.

The purpose of the study is to improve the efficiency of single-board computers with limited capabilities located at the edge layer of the Internet of Things. The characteristic features of the use of computers with limited computing capabilities in Internet of Things networks are determined. Reasoned choice of methods and algorithms for building a virtual cluster. The architecture of a virtual cluster of computers with limited resources is formed. An improved method of virtualization of the boundary layer of single-board computers of the Internet of Things due to the construction of a virtual cluster of heterogeneous devices with limited computing resources.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	8
ВСТУП .....	9
1 АНАЛІЗ ЗАСТОСУВАННЯ КОМП'ЮТЕРІВ З ОБМЕЖЕНИМИ ОБЧИСЛЮВАЛЬНИМИ МОЖЛИВОСТЯМИ У МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ .....	12
1.2 Аналіз використання одноплатних комп'ютерів в різних областях .....	13
1.3 Існуючі рішення формування кластера одноплатних комп'ютерів .....	16
1.4 Застосування кластерів на одноплатних комп'ютерах в мережах Інтернету речей .....	19
2 ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДІВ І АЛГОРИТМІВ ПОБУДОВИ ВІРТУАЛЬНОГО КЛАСТЕРУ ГРАНИЧНОГО ШАРУ ІНТЕРНЕТУ РЕЧЕЙ .....	21
2.1 Формування архітектури кластера одноплатних комп'ютерів .....	21
2.2 Вибір програмної платформи віртуалізації .....	24
2.3 Вибір алгоритму розподілених обчислень на кластері з одноплатних комп'ютерів .....	26
2.4 Вибір методу передачі повідомлень .....	27
2.5 Зберігання даних в кластері з одноплатних комп'ютерів .....	28
2.6 Алгоритми тестування продуктивності кластера .....	29
3 ПОБУДОВА ВІРТУАЛЬНОГО РОЗПОДІЛЕНОГО КЛАСТЕРА .....	31
3.1 Варіанти формування архітектури для віртуального кластера комп'ютерів з обмеженими ресурсами .....	31
3.2 Опис шарів архітектури віртуального кластера комп'ютерів з обмеженими ресурсами .....	34
3.2.1 Шар фізичних пристроїв .....	35
3.2.2 Шар формування і управління кластером .....	36

3.2.3 Координаційний шар кластера .....	39
3.2.4 Шар віртуального кластера .....	42
3.3 Етапи формування віртуального кластера .....	44
3.4 Алгоритми і методи формування віртуального кластера .....	47
3.5 Порівняння віртуального і класичного кластерів.....	52
3.6 Тестування віртуального кластера .....	53
3.7 Порівняльне тестування віртуального та класичного кластерів.....	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	66
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	68
ДОДАТОК Б Публікація .....	77
ДОДАТОК В Текст програм та запитів .....	79

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ  
І ТЕРМІНІВ

АВГК – архітектура віртуального гетерогенного кластера  
ВГВ – віртуальна група вузлів  
ВГП – віртуальна група пристроїв  
ВОП – віртуальні обчислювальні пристрої  
ВК – віртуальний кластер  
ІР – Інтернет речей  
КВ – Координаційний вузол  
КООР – комп'ютер з обмеженими обчислювальними ресурсами  
Мб – мегабайт  
МВ – Майстер-вузол  
ОВ – обчислювальний вузол  
ОЗП – оперативний запам'ятовуючий пристрій  
ОС – операційна система  
ПК – персональний комп'ютер  
ЕС – Edge Computing  
IoT – Internet of Things  
MMD – Mini-Micro Designer  
MPI – Message Passing Interface  
SBC – Single Board Computer  
VPN – Virtual Private Networks  
WAN – Wide Area Networks

## ВСТУП

У XXI сторіччі технології Інтернету речей все більше стають невід'ємною частиною нашого повсякденного життя, наприклад, це розумний будинок, розумне місто, сільське господарство, медицина, логістика тощо. Величезна кількість даних, генерованих пристроями Інтернету речей, вимагають подальшої постобробки і аналізу, в у тому числі з використанням методів машинного навчання. Для обробки інформації, що надходить, використовуються хмарні інфраструктури, де на потужностях дата-центру відбувається безпосередній аналіз та прийняття рішень на основі одержаних результатів. Однак, кількість підключених пристроїв до мережі Інтернет постійно збільшується, що призводить до складнощів хмарної обробки даних у зв'язку із збільшенням даних, що надходять, збільшенням навантаження на мережу, збільшенням затримки передачі між вузлами тощо. Не завжди допомагає і введення проміжного шару, наближеного до граничного шару Інтернету речей – туманних обчислень.

Одним з рішень даного питання є розвиток технологій граничних обчислень, що дозволяють взяти частину навантаження на себе і зменшити час відгуку на виникаючу подію. Для організації граничних обчислень останні 5-6 років почали використовувати одноплатні комп'ютери, які мають компактні розміри і високу енергоефективність, але малу продуктивність, що не дозволяє вирішувати обчислювально складні завдання з використання ресурсів одного пристрою. Одним із рішень даної проблеми є створення локального кластера, що складається з комп'ютерів з обмеженими обчислювальними ресурсами, які не містять додаткових обчислювальних розширювачів, співпроцесорів для обчислень спеціальних завдань. При цьому комп'ютери та кластери, задіяні у граничних обчисленнях, часто виконують певні завдання і не задіють всього потенціалу даних пристроїв, які хоч і є комп'ютерами з обмеженими обчислювальними ресурсами, але

можуть бути також використані для вирішення інших завдань залежно від вільних обчислювальних ресурсів. У рамках цієї дипломної роботи під віртуальним кластером будемо розуміється організовану певним чином множину комп'ютерів з обмеженими обчислювальними ресурсами для вирішення обчислювально складних завдань з використанням віртуалізації ресурсів доступних пристроїв в існуючій інфраструктурі мережі Інтернету речей. При цьому під віртуалізацією розуміється виділення частини вільних ресурсів пристроїв для вирішення завдань у рамках кластера. Актуальність цієї роботи обумовлена постійним збільшенням числа задіяних одноплатних комп'ютерів в мережах Інтернету речей, а також зростанням їх продуктивності, ресурси яких зазвичай не задіяні в повному обсязі, так як вирішують строго певну низку завдань. Ці потужності можна використовувати для побудови віртуального кластера на основі існуючої інфраструктури для вирішення цілого ряду інших завдань, наприклад, таких як:

- аналіз даних від кінцевих пристроїв в області граничних обчислень для перерозподілу обчислення навантаження на дата-центри з використанням вже існуючої інфраструктури;

- використання вільних обчислювальних ресурсів для вирішення обчислювально складних завдань на основі пристроїв граничного шару без необхідності придбання додаткового обладнання або залучення потужностей туманного або хмарного шарів;

- організація обчислювальної групи пристроїв для формування шару постійної пам'яті на основі вільних ресурсів Інтернету речей;

- віртуалізація використання обчислювальних ресурсів мережі для розподілу та зберігання даних на пристроях граничного шару Інтернету речей.

Отже, спираючись на вищевикладене, науково-технічне завдання, спрямоване на розробку методу віртуалізації граничного шару одноплатних комп'ютерів Інтернету речей є актуальним та доцільним.

Предметом дослідження є моделі та методи віртуалізації граничного шару одноплатних комп'ютерів Інтернету речей.

Об'єктом дослідження є процес організації віртуального кластеру пристроїв граничного шару одноплатних комп'ютерів Інтернету речей.

Метою дослідження є підвищення ефективності використання одноплатних комп'ютерів з обмеженими можливостями, розташованих на граничному шарі Інтернету речей.

Для досягнення мети повинні бути вирішені такі часткові завдання:

1) провести аналіз застосування комп'ютерів з обмеженими обчислювальними можливостями у мережах Інтернету речей;

2) обґрунтувати вибір методів і алгоритмів побудови віртуального кластеру;

3) сформуванати архітектуру віртуального кластера комп'ютерів з обмеженими ресурсами;

4) побудувати та дослідити ефективність віртуального розподіленого кластера граничного шару одноплатних комп'ютерів Інтернету речей.

Наукова новизна дослідження полягає в такому:

удосконалений метод віртуалізації граничного шару одноплатних комп'ютерів Інтернету речей за рахунок побудови віртуального кластеру гетерогенних пристроїв з обмеженими обчислювальними ресурсами, що надало вигаш за часом виконання завдань в середньому на 8%.

Практична цінність дослідження полягає в підвищенні продуктивності використання пристроїв з обмеженими обчислювальними ресурсами на граничному шарі Інтернету речей.

# 1 АНАЛІЗ ЗАСТОСУВАННЯ КОМП'ЮТЕРІВ З ОБМЕЖЕНИМИ ОБЧИСЛЮВАЛЬНИМИ МОЖЛИВОСТЯМИ У МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ

## 1.1 Інтернет речей в сучасному світі

Інтернет речей (Internet of Things, IoT) є однією з тих областей, яка все більше входить в наше повсякденне життя та демонструє швидкий зріст технологій. На даний момент кількість підключених пристроїв до мережі Інтернет вже більше ніж в 2 рази перевищило населення Землі, а по прогнозах в 2030 році кількість підключених пристроїв повинно було скласти понад 29 мільярдів [1]. Світовий ринок Інтернету речей у 2023 році склав 374,49 млрд доларів та очікується досягнення 1 562 млрд доларів у 2032 році (рисунок 1.1).

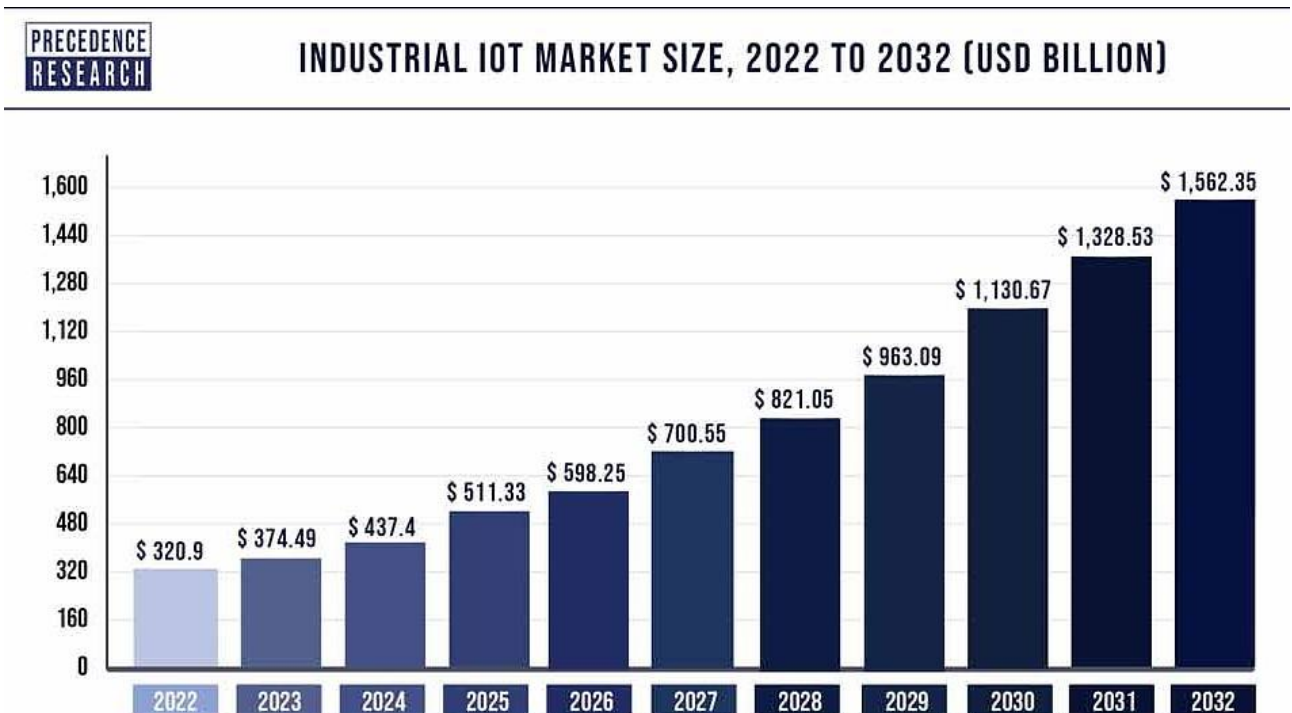


Рисунок 1.1 – Зростання ринку Інтернету речей [2]

Однак глобальна епідемія Covid-19 загальмувала кількісне зростання

пристроїв Інтернету речей і направила дослідників на вивчення і адаптацію існуючих рішень для виявлення і контролю нових захворювань [3], а також розширення використання пристроїв з обмеженими обчислювальними ресурсами в граничних обчислень (“Edge Computing”, EC) [4], розвитком комунікації, наприклад, нових мереж мобільного зв’язу 5G та 6G [5] тощо.

У індустріальній сфері все більше застосовуються граничні обчислення, що дозволяють знизити навантаження на хмари та зменшити трафік у мережі [6]. З появою одноплатних комп’ютерів активізувалися дослідження, спрямовані на застосування даних пристроїв в граничних обчислення з метою мінімізації реагування системи на події [7]. Але одноплатні комп’ютери знаходять у останні роки все більше застосування не тільки як пристрої граничних обчислень, але й у різних областях, що детальніше розглянуто у наступному підрозділі.

## 1.2 Аналіз використання одноплатних комп’ютерів в різних областях

Одноплатний комп’ютер (Single Board Computer, SBC) – самодостатній комп’ютер, зібраний на одній друкований платі, на якій встановлені всі необхідні компоненти для забезпечення його функціонування: процесор, оперативна пам’ять, системи введення-виведення і т.д. Перший, по-справжньому одноплатний комп’ютер, з’явився ще в 1976 році і називався “MMD-1” (Mini-Micro Designer 1), але при цьому по-справжньому масовим і доступним даний формат виконання комп’ютерів став лише тільки у 2012 році з появою маленького одноплатного комп’ютера Raspberry Pi, спрямованого на вивчення базових комп’ютерних навичок школярами, тобто націленого в першу чергу на освітню сферу і навчання користування комп’ютером, програмування тощо [8].

Маючи низьку ціну, маленький розмір (площа розташування приблизно розміром з банківську карту, рисунок 1.2) і достатню продуктивність для запуску повноцінної операційної системи на базі Linux він здобув свою

популярність у багатьох ентузіастів та дослідників з різних областей.

Завдяки успіху Raspberry Pi інші виробники стали пропонувати свої моделі одноплатних комп'ютерів і на поточний момент в світі існує більше 200 моделей одноплатних комп'ютерів.

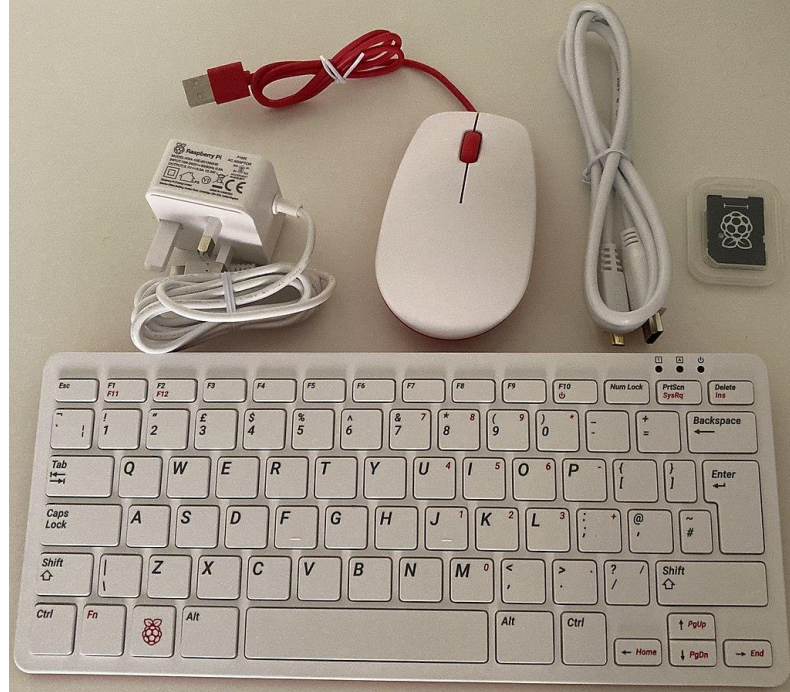


Рисунок 1.2 – Комплект поставки моделі Raspberry Pi 400

Найбільш популярними є різні моделі Raspberry Pi 3 і 3b+, RP Zero W та модель 2020 року Raspberry Pi 400, яка має кілька варіантів виконання з 2 ГБ, 4 ГБ і 8 ГБ ОЗП. З інших виробників стоїть відзначити Banana Pi M5, RockPro64, Odroid N2. Порівняльні характеристики одноплатних комп'ютерів представлені в таблиці 1.1.

Багато виробників одноплатних комп'ютерів пішли шляхом випуску модульних плат у форматі підключення SO-DIMM, прикладом може бути обчислювальний модуль Raspberry Pi3+, відмінною особливістю якого виступає сховище даних в виконанні eMMC накопичувача з 32ГБ постійною пам'яті, на відміну від молодшої моделі з micro-SD карткою. Це рішення в першу чергу розраховане на індустріальний сектор.

Таблиця 1.1 – Порівняння характеристик одноплатних комп'ютерів

Модель	RAM, GB	SoC	GPU	Ethernet	Сховище	Ціна
Raspberry Pi 3b+	1 LPDDR2	ARM A53 Cortex	VideoCore IV	330 Mbit, 2.4 GHz, 5GHz Bluetooth 4.2	microSD – карти	35\$
Raspberry Pi 400	2, 4, 8 LPDDR4	4 x Cortex- A72	VideoCore VI	1000 Mbit/s 802.11b / g / b / ac WiFi 5 Bluetooth 5.0	microSD – карти	47 \$, 74 \$, 103 \$
RP Zero W	0.5	1 X ARM1176JZF S	VideoCore IV	802.11 b/g/n wireless LAN BLE 4.1	microSD – карти	10 \$
RockPro64	4 LPDDR4	4 x ARM CortexA53	Mali - T860-MP4	1000 Mbit /s Ethernet	слот для модуля eMMC	79\$
Banana Pi M5	4 GB LPDDR4	Quad-Core Cortex-A55	Mali-G31 GPU	1000 Ethernet Mbit /s	microSD – карти, слот для eMMC	53 \$
Odroid N2	2, 4	Quad-core Cortex-A73 Dual-core Cortex-A53	Mali-G52 GPU	1000 Mbit /s Ethernet Optional WiFi USB adapters	microSD – карти, слот для eMMC	73\$ - 83\$

У кінці 2020 р. і початку 2021 р. почали пров'яляться плати в новому модульному рішенні, прикладом такої плати є одноплатний комп'ютер Raspberry Pi Compute Module 4. Як і попередня модель, даний модуль має тільки модуль підключення до основної плати, який був замінений з SO-DIMM виконання на користь 100-контактних роз'ємів Hirose DF40C-100DS-0.4V, що дозволило збільшити швидкість обміну даними між платами та зменшити розміри самої плати. Плати такого формату насамперед розраховані на розробників і індустріальний сектор та знайшли своє застосування в сфері Інтернету речей, зокрема в області граничних обчислень. Також плата підходить для створення IoT-проектів різної складності. Але у багатьох випадках такі плати об'єднуються в кластери, що буде розглянуто у наступному підрозділі.

### 1.3 Існуючі рішення формування кластера одноплатних комп'ютерів

Одноплатні комп'ютери відносять до пристроїв з обмеженими обчислювальними ресурсами, тому що їх продуктивність та фізичні ресурси пристроїв не дозволяють вирішувати обчислювально складні завдання і для рішення цієї проблеми їх об'єднують в кластери. До недавнього часу кластерні обчислення були надто дорогими та складними для звичайних користувачів, однак популярність одноплатних комп'ютерів привела до появи дешевих кластерів, заснованих на цих пристроях.

Також одноплатні комп'ютери та кластери з одноплатних комп'ютерів знайшли своє застосування у концепції граничних обчислень, дозволяючи реалізувати оптимізацію хмарних обчислень шляхом розміщення обчислювальних ресурсів якомога ближче до джерела (кордон мережі), що підвищує оперативність обробки інформації з кінцевих пристроїв. Наступним кроком розвитку цієї концепції вважається реалізація надграничних обчислень (Extreme edge), прикладом може виступати технологія «smart dust» («розумна пил»), побудована на застосування бездротових самоорганізованих пристроїв з метою обробки та передачі даних в системі Інтернету речей.

Слід зазначити, що з 2016 року ведуться дослідження з розробки різних методів і інструментів для організації обчислень і розподілених сховищ даних на одноплатних комп'ютерах в граничних обчисленнях, але на даний момент готових рішень в цієї області немає. Але спроби організації розподілених обчислень на «доступних» обчислювальних системах робилися й раніше. Так у 2007 році Гурав Кханна з кафедри фізики Масачусетського університету зібрав свій перший кластер з 16 ігрових приставок PlayStation 3 для моделювання сутічок чорних дір, а в 2014 року кластер налічував вже 412 приставок.

У 2010 року з метою створення дешевого і продуктивного суперкомп'ютера комп'ютера у порівнянні з аналогами ВПС США побудували систему, що включає у собі 1760 пристроїв PlayStation 3,

назвавши суперкомп'ютер « Condor Cluster », продуктивність системи склала 500 TFLOPS. Кластер складається з 168 окремих графічних процесорів та 84 координуючих серверів для направлення трафіку всередині системи. Рішення створити кластер на основі ігрових приставок було прийнято на основі економічних показників, тому що кожна приставка на момент створення кластера коштувала приблизно 400\$, при цьому порівнянна технологія коштувала б близько 10000 \$ за одиницю. Додатково відзначається енергоефективність системи, яка складає приблизно 10% потужності аналогічного суперкомп'ютера. Але через проблеми з підтримкою програмного забезпечення та закриття можливості встановлення сторонньої операційної системи (для всіх пристроїв) Sony проєкт було закрито.

Незважаючи на те, що Playstation 3 не є одноплатним комп'ютером, однак напрацювання, використовувані для організації суперкомп'ютера на основі окремих самостійних пристроїв використовуються для організації кластерних систем з одноплатними комп'ютерами. Стоїть відзначити, що продуктивність приставки Playstation 3 на CPU складала приблизно 230 GFLOPS в 2006 році, а споживання електрики в режимі бездіяльності 150 Вт при навантаженні 200-204 Вт, що дає енергоефективності продуктивність приблизно 1,15 GFLOPS на 1 Вт електроенергії. При цьому один із самих продуктивних комп'ютерів на 2020 рік Raspberry Pi 4 показує продуктивність в CPU-тесті 9.92 GFLOPS, а споживання пристрою в режимі бездіяльності 2,47 Вт та при максимальному навантаженні 6,79 Вт, що становить 1,46 GFLOPS на 1 Вт. Таким чином одноплатний комп'ютер демонструє ще більшу енергоефективність у порівнянні із PlayStation 3 та потенційно кластери на одноплатних комп'ютерах можуть демонструвати ще більш високу енергоефективність в обчислювальних завданнях при рівній продуктивності.

З 2012 року почалося масове використання одноплатних комп'ютерів у дослідній діяльності, а також у розробці власних проєктів та проведенні експериментів.

Як зазначалося раніше, з 2016 року особливу увагу приділено дослідженням застосування одноплатних комп'ютерів в граничних обчисленнях систем Інтернету речей. Деякі проекти створення кластерів та їх застосування в різних областях наведено в таблиці 1.2. Представлені в цій таблиці роботи показують лише деякі напрацювання які ведуться в напрямку дослідження і застосування одноплатних комп'ютерів, їх важливість та перспективному у майбутньому складно переоцінити.

Таблиця 1.2 – Огляд областей застосування одноплатних комп'ютерів

Область застосування	Використовувані пристрої	Виконувані завдання	Рік
У освітніх цілях	Raspberry Pi	Виконання обчислювальних задач; тестування продуктивності роботи системи в різних фреймворках, в тому числі Hadoop	2013
		Тестування продуктивності роботи системи в різних завданнях	2015
		Оцінка застосування одноплатних комп'ютерів для здійснення кібероперацій	2016
	Raspberry Pi B	Навчання кластерним обчисленням	2014
	Raspberry Pi 2	Виконання обчислювальних задач, навчання кластерним обчисленням	2017
		Виконання обчислювальних завдань, моделювання і прогнозування	2018
	Raspberry Pi 3	Виконання обчислювальних задач, демонстрація можливостей, навчання кластерним обчисленням	2020
		Робота з великими даними, аналіз продуктивності розподіленою системи Hadoop	2022
Граничні обчислення (edge computing)		Хмарні обчислення в концепції граничних обчислень	2016
	Raspberry Pi	Застосування кластера в якості обчислювального вузла для граничних обчислень, оцінка продуктивності контейнерних служб	2017
	Pi Stack	Управління віддаленими одноплатними комп'ютерами на базі Beowulf	2018
	Raspberry Pi Odroid XU4	Розподілена потокова обробка даних, управління кінцевими пристроями, потокова обробка даних	2019
	RockPro64	Розподілена глибока нейронна мережа на кластері з одноплатних комп'ютерів	2021

Представлені в таблиці 1.2 роботи показують лише деякі напрацювання, які ведуть в напрямку дослідження і застосування одноплатних комп'ютерів, їх важливість та перспективність у майбутньому складно переоцінити. Незважаючи на скромну продуктивність одноплатні комп'ютери вже зараз знайшли своє застосування в багатьох галузях промисловості, особливий відгук отримала сфера Інтернету речей і граничних обчислень. Однак усі напрацювання і спроби реалізації високопродуктивних обчислень на такому кластері носять експериментальний характер та готового рішення наразі немає.

Яким чином одноплатні комп'ютери використовуються в Інтернеті речей буде розглянуто у наступному підрозділі.

#### 1.4 Застосування кластерів на одноплатних комп'ютерах в мережах Інтернету речей

Кластери з одноплатних комп'ютерів довели свою ефективність не тільки в сфері освіти і навчання роботі з кластерними системами, але і їх високу ефективність застосування в промисловості для контролю різних систем, також вони знайшли своє застосування в робототехніці, в «розумних системах», наприклад: паркування, розумні дома, вуличне освітлення. Кластерні системи на основі одноплатних комп'ютерів застосовуються для підтримки веб-сервісів і різних серверів, так як така система здатна ефективно масштабуватись та підключати додаткові вузли по мірі зростання навантаження на систему, таким чином забезпечуючи енергоефективність роботи комплексу загалом. При цьому якщо один із вузлів вийде з ладу, то гарантується відмовостійкість системи, а низька вартість одноплатного комп'ютера у порівнянні із звичайною системою суттєво зменшує витрати на обслуговування.

Кількість задіяних одноплатних комп'ютерів постійно зростає і надалі цей зріст буде тільки збільшуватися, при цьому доступні ресурси

одноплатних комп'ютерів або класичних кластерних систем не завжди використовуються повністю або зовсім простоюють.

Ідея «віртуального кластера» (далі ВК) полягає в об'єднанні і організації розрізнених гетерогенних пристроїв для вирішення різних складних обчислювальних завдань з використанням вільних (доступних) ресурсів вже існуючою інфраструктурою, що знаходяться в області граничних обчислень, насамперед, розрахованою на використання ресурсів одноплатних комп'ютерів. Такий кластер також дозволить використовувати ресурси існуючої інфраструктури раціональніше, наприклад, розгорнувши додаткові послуги з обробки та зберігання даних.

Отже, розробка методу побудови віртуального кластеру одноплатних комп'ютерів, що розташовані на граничному шарі системи Інтернету речей, є своєчасною та актуальною.

## 2 ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДІВ І АЛГОРИТМІВ ПОБУДОВИ ВІРТУАЛЬНОГО КЛАСТЕРУ ГРАНИЧНОГО ШАРУ ІНТЕРНЕТУ РЕЧЕЙ

### 2.1 Формування архітектури кластера одноплатних комп'ютерів

В основному кластери збирають на пристроях однієї моделі, тому що це дозволяє спростити подальше його обслуговування, вважаючи на єдине програмне забезпечення і обмежені технічні ресурси одноплатних комп'ютерів. Стандартна архітектура кластера представляє собою виділений керуючий Майстер-вузол (МВ), який має роль «лідера» і обчислювальні вузли (ОВ). Усе пристрої, з'єднані за допомогою комутатора. На рисунок 2.1 представлений приклад реалізації такого кластера з використанням шести одноплатних комп'ютерів.

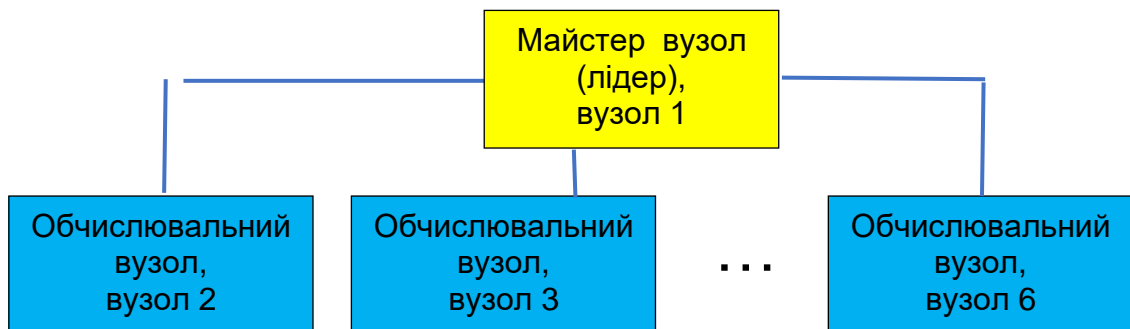


Рисунок 2.1 Типова архітектура для кластера з одноплатних комп'ютерів

Оскільки ця архітектура є «еталонною», існує величезна кількість рішень від аматорських, де можна, можливо самостійно зібрати свій власною кластер по інструкції, до комерційних з вже зібраної і налаштованою системою по бажанню користувача, наприклад Pico Cluster (рисунок 2.2).

Індустріальний сектор зацікавився компактними кластерами на основі одноплатних комп'ютерів, але розглянуте раніше виконання занадто громіздке для використання і більше підходить в дослідницьких цілях. Тому були розроблені і випущені плати що дозволяють розміщувати на собі

відповідні модулі для створення кластера в компактних розмірах, наприклад SOPINE Clusterboard від компанії Pine для семимодулів SOPINE A64 при цьому має розміри стандартної m- itx плати (рисунок 2.3).



Рисунок 2.2 – Pico Cluster на основі п'яти Raspberry Pi 4 [8]



Рисунок 2.3 – SOPINE Clusterboard плата з 7 модулями SOPINE A64 [9]

Проте обмежені ресурси одноплатних комп'ютерів не дозволяють

«нескінченно» збільшувати кількість робітників вузлів, тому при нестачі ресурсів Майстер вузла необхідно додати в систему Координаційний вузол (КВ), котрий буде отримувати завдання від майстер вузла та видавати їх підпорядкованим обчислювальним вузлам, при цьому в випадку виходу основного МВ його місце може зайняти КВ згідно заздалегідь виданому пріоритету, тим самим досягається відмовостійкість сервісу (рисунок 2.4).

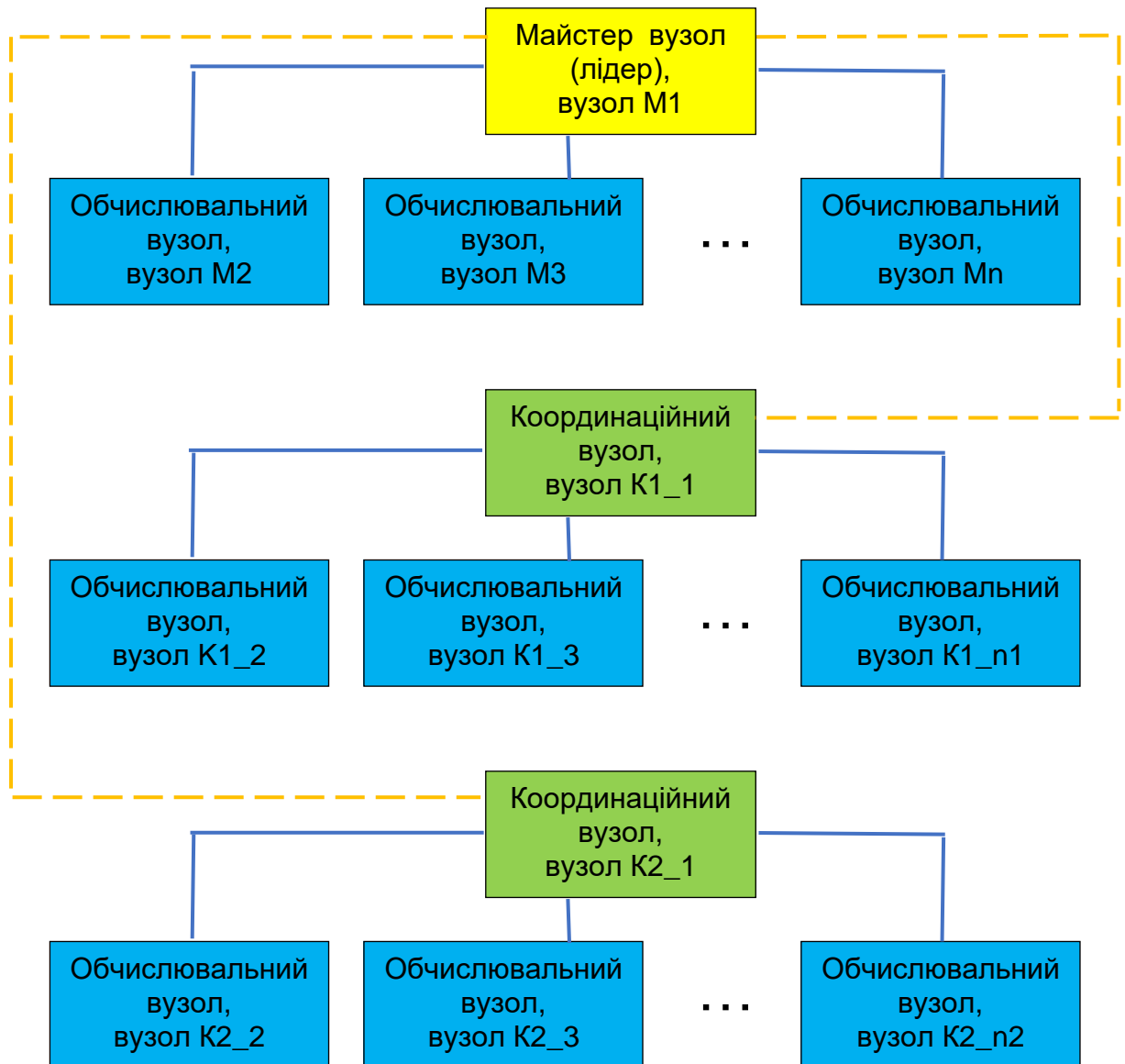


Рисунок 2.4 – Архітектура кластера з координаційними вузлами



а



б

Рисунок 2.5 – Кластер Beast (а) та кластер від Oracle (б)

Прикладом даної архітектури виступають експериментальні кластери, зібрані для освітній і дослідницької діяльності, наприклад як кластер Beast (рисунок 2.5, а) або з метою демонстрації технологій та можливості від Oracle на прикладі 1060 плат Raspberry Pi3b+ (рисунок 2.5, б).

## 2.2 Вибір програмної платформи віртуалізації

Одним із можливих рішень для віртуалізації може виступати продукт компанії VMware vSphere, це програмне забезпечення для встановлення та управління віртуальними машинами. Це рішення дозволяє віртуалізувати ресурси хоста і задіяти їх для рішення поставленої завдання через взаємодію з віртуальною машиною. Основним недоліком такого рішення є надзвичайно високі системні вимоги до кінцевого пристрою, порівняно з тим, що може надати одноплатний комп'ютер, другим мінусом є пропріетарність цього рішення (обмеженість прав при використанні).

Компанія VMware випустила в 2022 року остаточну версію продукту ESXi з можливістю його запуску на одноплатному комп'ютері Raspberry Pi. Для цього їм довелося видалити безліч служб і переробити архітектуру, але зрештою у версії «тільки для розробників» з'явилася можливість запустити гіпервізор на одноплатному комп'ютері. Однак, згідно із заявами користувачів, дана розробка є середовищем тестування технологій на ARM процесорах і не призначена для повноцінної роботи, також є пропрієтарною системою, а значить не підходить для рішення розглядаємого завдання.

Docker являє собою відкриту платформу для розробки, доставки та запуску застосунків. Технологія Swarm дозволяє ініціювати роботу з системою підлеглих вузлів, але основним її недоліком є складність масштабування системи та підтримка системи спрямована у бік продуктивних рішень. Існуючі Docker Swarm кластери на одноплатних комп'ютерах працюють не стабільно і цікаві тільки в освітніх цілях. Однак є спроби використовувати кластер на одноплатних комп'ютерах, так компанія Pine використовує кластер на RockPro64 для роботи веб-сервера і інших сервісів (рисунок 2.6, [12]).

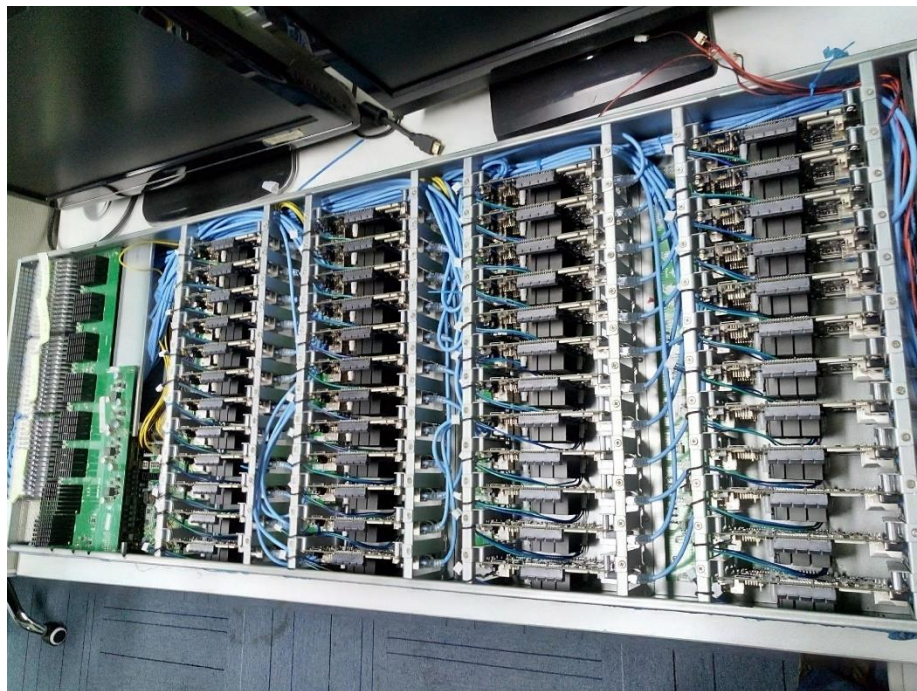


Рисунок 2.6 – Кластер із 48 вузлів RockPro64 [12]

Принципова відмінність Kubernetes від Docker полягає в тому, що Kubernetes призначений для роботи через кластер, а Docker запускається тільки на одному вузлі. Ці технології краще всього працюють спільно, Kubernetes дозволяє керувати пулом віртуальних машин, розподіляючи навантаження між ними, в том числі при необхідності задіяти вільні вузли. На відміну від Docker це програмне забезпечення має ще великі системні вимоги та може призводити до збоїв в роботі кластера.

### 2.3 Вибір алгоритму розподілених обчислень на кластері з одноплатних комп'ютерів

Основними вимогами при проведенні вибору алгоритму розподілених обчислень на кластері з одноплатних комп'ютерів є, по-перше, можливість його реалізації на такому комп'ютері, а, по-друге, наявність балансувальника навантаження між вузлами.

Основний елемент балансування навантаження – це застосовуваний алгоритм. Завдання алгоритму складаються в такому:

- розподіл завдань між вузлами (усі вузли повинні бути завантажені завжди);
- забезпечення високої швидкості обробки запитів;
- необхідність своєчасного оброблення кожного запиту.

Розглянемо найбільш популярні алгоритми балансування навантаження.

Алгоритм Round Robin (RR) використовується у більшості обчислювальних систем. Суть алгоритму полягає в упорядкуванні елементів або завдань і їх розподіл за круговим циклом, на кожне завдання виділяється певний інтервал часу для його виконання (квант часу), якщо завдання не виконане, то воно повертається до черги.

Даний алгоритм є самим популярним алгоритмом, що

використовується в обчислювальних системах, основним недоліком є необхідність підбору оптимального інтервалу часу, що є дуже складним завданням.

Алгоритм *Weighted Round Robin* є вдосконаленим алгоритмом *Round Robin*. У цьому алгоритмі вводяться вагові коефіцієнти, які присвоюються відповідному вузлу відповідно з його параметрами (насамперед, продуктивністю). Алгоритм добре підходить для малих локальних мереж.

Алгоритм *Sticky Sessions* використовується у випадку, при якому запити клієнта передаються на один і той ж сервер групи, тобто запити розподіляються на основі IP-адреси клієнта. Даний метод гарантує передачу запитів одного і того ж клієнта на той ж сервер. Недолік використання даного алгоритму з'являється при використанні клієнтом динамічної IP-адреси або проксі сервера, рішенням виступає використання *cookie*-файлів. Також у випадку відключення сервера всі поточні сесії губляться.

Основна особливість алгоритму *min-max* полягає в розподілі завдань на вузли за таким принципом: мінімальний час виконання максимально великих завдань. Алгоритм добре підходить для гетерогенних систем, так як вузлам задаються ваги з часом виконання завдань ( $C_{ij}$ ), які розраховується згідно з формулою:

$$C_{ij} = E_{ij} + r_j, \quad (2.1)$$

де  $E_{ij}$  – час виконання завдання  $i$  на вузлі  $j$ ;  $r_j$  – час відгуку вузла  $j$ .

Зворотний алгоритм (*min-min*) працює за методом відправлення мінімального завдання на вузол з мінімальним часом виконання.

## 2.4 Вибір методу передачі повідомлень

Існує багато способів організувати обмін повідомленнями між пристроями, розглянемо деякі з них:

Socket (Сокет) – програмний інтерфейс для забезпечення обміну даними між процесами. Простий спосіб організації передачі даних, на серверній машині створюються порти, до яких підключаються клієнтські машини передачі даних. Такий спосіб добре підходить для централізованих систем.

Message Passing Interface (MPI, інтерфейс передачі повідомлень) – інтерфейс що дозволяє організувати передачу даних по API в паралельному програмуванні. Даний інтерфейс використовується при розробці програм для кластерів і суперкомп'ютерів. Існує велика кількість бібліотек з різної реалізацією MPI, але основним недоліком більшості реалізацій є складність роботи з комп'ютерами, які не належать до однієї мережі.

REST API – уніфікований інтерфейс передачі даних для взаємодії розподілених застосунків в мережі. Використовується для організації обміну повідомленнями вебслужб. Брокери повідомлень дозволяють організувати швидкий обмін повідомленнями, однак даний спосіб має суттєвий недолік, виражений в неоптимальній роботі при передачі файлів, що призводить до обмежень по розміру повідомлень, що передаються.

## 2.5 Зберігання даних в кластері з одноплатних комп'ютерів

У звичайних кластерах за зберігання відповідають спеціальні високопродуктивні системи, що забезпечують високу швидкість читання та запису файлів. У кластері із одноплатних комп'ютерів дані найчастіше зберігаються на Майстер вузлі, постійна пам'ять якого представлена microSD картою з низькою швидкістю роботи з даними та рідше eMMC пам'яттю. У системах, що розглядаються, модуль постійної пам'яті часто є вузьким місцем, яке обмежує продуктивність системи. Для рішення даної проблеми використовується зовнішнє сховище даних, на якому зберігаються файли для обробки. Сховищем даних може бути представлений usb накопичувач, підключений до майстер вузлу. В такому випадку обмеження швидкості

роботи з даними залежить від специфікації USB порту пристрою, або на мережному диску/хмарі, у такому разі робота з даними обмежується можливістю мережевої взаємодії пристроїв.

## 2.6 Алгоритми тестування продуктивності кластера

При мережній взаємодії двох вузлів між собою основними мережевими характеристиками є латентність і пропускна здатність. Якщо пропускна здатність в першу черга обмежується мережним інтерфейсом і шириною каналу передачі даних, то латентність (затримка) складається з часу, який витрачається програмним забезпеченням та пристроєм мережі на підготовку до передачі повідомлення.

Латентність розраховується за такою формулою:

$$S = \frac{T}{2N}, \quad (2.2)$$

де  $T$  - число ітерацій у кожній тестовій процедурі;  $N$  - кількість повторів операцій.

Пропускна здатність між пристроями заміряється за такою формулою:

$$K = \frac{2 \cdot N1 \cdot L}{t}, \quad (2.2)$$

де  $L$  – довжина повідомлення в байтах,  $N1$  – кількість повторень з метою мінімізації похибки за рахунок усереднення результату,  $t$  – час, витрачений на обмін даними.

Тестування швидкості розрахунку матриць. Додавання матриць є простим і ефективним способом обчислення продуктивності кластера при нестандартних конфігурації, що дозволяє порівняти час виконання одного завдання на різних системах.

Розглянемо алгоритм додавання матриць  $A$  і  $B$  розмірністю  $m \times n$

на обчислювальному кластері з кількістю вузлів  $k$ . Для цього необхідно виконати розбиття матриць  $A$  і  $B$  на підматриці в залежності від  $m$ ,  $n$  і  $k$ .

У випадку, коли  $m \cdot n \leq k$ , розбиття тривіально. Для зручності матриці  $A$  та  $B$  можна згорнути в одновимірні вектори  $A$  і  $B$ , де  $A = (a_{1,1}, \dots, a_{m,n})$ ,  $B = (b_{1,1}, \dots, b_{m,n})$ . Тоді на кожен  $i$ -й обчислювальний вузол, де  $i = 1, \dots, m, n$ , припадає по елементу  $a$  і  $b$  із відповідних векторів. Таким чином, кожен  $i$ -й обчислювальний вузол робить додавання скалярних значень  $a$  і  $b$ .

Якщо  $m \cdot n > k$ , необхідно поділити кожен з матриць на  $k$  підматриць. Таким чином, кожен  $i$ -й обчислювальний вузол здійснює додавання двох матриць, по можливості застосовуючи власні апаратні оптимізації щодо розпаралелювання даної операції.

Після завершення додавання, результуюча матриця збирається з підматриць відповідно із зробленим раніше розбиттям.

Тестування розпізнавання зображення з використанням згорткової нейронної мережі. Розпізнавання зображень є одним із складних обчислювальних завдань, вимагає суттєвих обчислювальних ресурсів пристроїв. Здебільшого для роботи із зображенням використовують графічні пристрої (відеокарти), але не на всіх одноплатних комп'ютерах вони є, а якщо є, то не задовольняють вимогам існуючих бібліотек. Тому тестування буде проводитися з використанням CPU доступних пристроїв з бібліотекою «face detector», заснованою на моделі згорткової нейронної мережі, яка має готові приклади і передбачену модель для множини зображень, що дозволяє використовувати даний тест як бенчмарк. У даному тесті використовуються зображення розміром 1000 x 500 пікселів та ємністю 1,5-2 МБ.

## 3 ПОБУДОВА ВІРТУАЛЬНОГО РОЗПОДІЛЕНОГО КЛАСТЕРА

### 3.1 Варіанти формування архітектури для віртуального кластера комп'ютерів з обмеженими ресурсами

У більшості випадків архітектура Інтернету речей представлені в вигляді шаруватої архітектури (архітектури по рівнях) і в залежності від завдання кількість шарів може змінюватись від трьох до семи. Розглянемо приклад тришарової архітектури, яка складається з трьох таких рівнів (рисунок 3.1, а):

- на рівні сприйняття (Reception Layer) знаходяться пристрої, які мають датчики, відповідальні за отримання інформації, і виконавчі механізми для взаємодії з довкіллям;

- мережний рівень (Network Layer) відповідає за взаємодію пристроїв з іншими мережними пристроями та серверами;

- програмний рівень (Application Layer) відповідає за доставку «послуг» кінцевому користувачеві, на цьому рівні розташовуються сервера, туманне та хмарне середовища.

Одним із важливих моментів трирівневої архітектури є той факт, що у ній немає рівня для бізнесу. Для вирішення цієї проблеми запропонована інша архітектура, яка є розширенням тришарової і включає в себе п'ять шарів з введенням додаткових рівнів: обробки (Processing Layer) та бізнес (Business Layer), яка наведена на рисунок 3.1, б. У цій архітектурі рівень «Network Layer» відповідає за передачу даних від пристроїв на рівень «Processing Layer», який відповідає за зберігання, аналіз і обробку даних, отриманих з різних пристроїв.

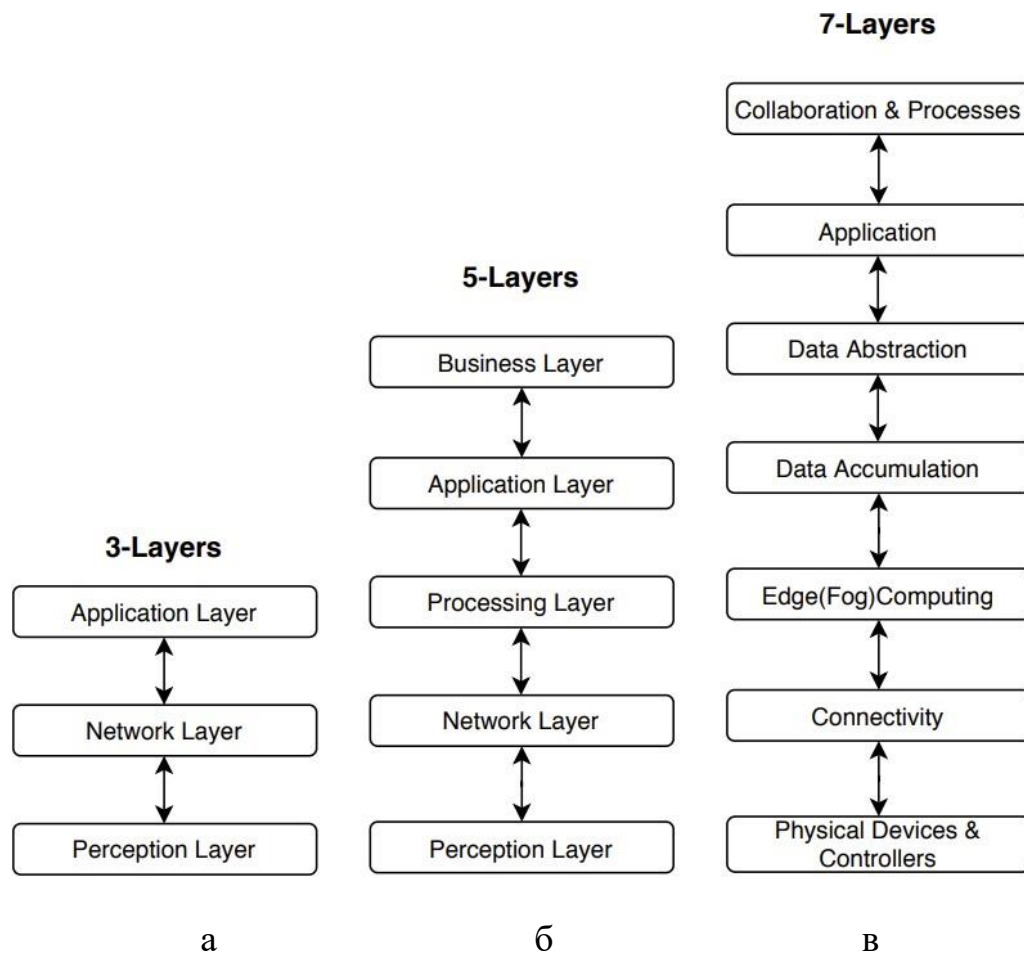


Рисунок 3.1 – Рівні архітектури Інтернету речей

На цьому шарі можуть застосовуватись різні технології, наприклад, туманні або хмарні обчислення та модулі обробки даних. Останній, «Business Layer» – це рівень управління системою Інтернету речей, він включає моделі застосунків, бізнесу і розрахунку прибутку, а також відповідає за конфіденційність користувачів.

Найбільш універсальна, семирівнева архітектура була запропонована на всесвітньому форумі Інтернету речей у 2018 року (Internet of Things World Forum (IoTWF)), яка наведена на рисунок 3.1, в. Розглянемо кожен шар даної архітектури:

- на першому рівні розташовані різні пристрої, датчики і контролери, що забезпечують їх взаємозв'язок;
- другий рівень відповідає за усі підключення і передачу даних в системі Інтернету речей;

- третій рівень, це шар граничних/туманних обчислень, на ньому проводиться аналіз і перетворення даних;
- четвертий рівень відповідає за накопичення даних, а також взаємодію з базою даних, а також гарантує правильність руху/потоків даних.
- на п'ятому рівні дані готуються для аналізу з використанням методів інтелектуального аналізу даних;
- шостий рівень – шар застосунків, де користувачі можуть використовувати інформацію з пристроїв Інтернету речей;
- сьомий рівень – шар є набором методів, які використовують дані для прийняття рішень на основі даних, отриманих з екосистеми Інтернету речей.

Є і інші варіанти побудови архітектури, наприклад, чотиришарова архітектура Windows Azure, представлена на рисунку 3.2. У даній архітектурі рівень «Application Layer» об'єднує в собі не тільки різні фізичні пристрої, а й програмне забезпечення. Рівень застосунків відповідає за обробку даних і прийняття рішень, тобто включає шар бізнесу з п'ятишарової архітектури. Шар даних винесений окремо і відповідає безпосередньо за зберігання, синхронізацію та подання даних.

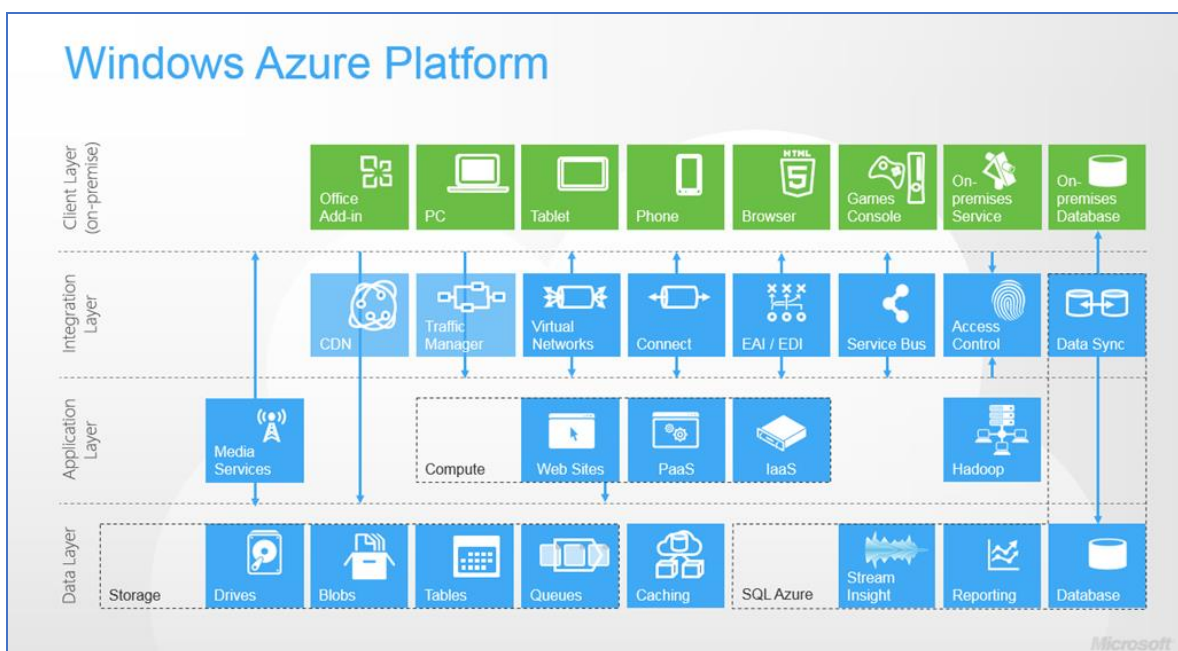


Рисунок 3.2 – Архітектура Windows Azure [14]

### 3.2 Опис шарів архітектури віртуального кластера комп'ютерів з обмеженими ресурсами

Архітектура для віртуального кластера припускає використання комп'ютерів з обмеженими обчислювальними ресурсами, які не однорідні за своїми характеристиками, що потребує вирішення задачі підбору подібних за характеристиками пристроїв та формування з них віртуального кластера. Так як кластерів може бути сформовано декілька, то впливає завдання розробки шару управління ними, а також надання дискового простору для зберігання файлів на базі комп'ютерів з обмеженими обчислювальними ресурсами. Як і у раніше розглянутих прикладах архітектури кластерів Інтернету речей в розробленій архітектурі для віртуального кластера застосовуються мікросервіси, котрі припускають використання для кожного типу даних окремого мікросервісу, незалежного від решти інших, що дозволяє шляхом заміни модуля забезпечити адаптивність і конфігурованість системи прийому різних типів даних. Основною особливістю є формування всіх шарів кластера на основі підключаємих зовнішніх пристроїв та вибір конфігурації за заданими параметрами для забезпечення виконання обчислювального завдання.

Отже, зупинимось на тому, що розподілена архітектура віртуального гетерогенного кластера (АВГК) повинна включати чотири шари (рисунок 3.3):

- шар фізичних пристроїв;
- шар формування та управління кластером;
- шар координації кластера;
- шар віртуальної групи пристроїв кластера.



Рисунок 3.3 – Архітектура віртуального гетерогенного кластера

При цьому слід відзначити головну особливість комп'ютерів з обмеженими обчислювальними ресурсами – це нестача ресурсів одного, окремо взятого комп'ютера для вирішення обчислювально складного завдання. Тому розглядається масштабування параметра зберігання даних (постійної пам'яті) засобами кількох комп'ютерів для забезпечення достатнього обсягу для зберігання оброблюваних файлів та отриманих результатів на координаційному шарі віртуального кластера.

Розглянемо дані шари віртуального кластера більш детально.

### 3.2.1 Шар фізичних пристроїв

На даному шарі представлені гетерогенні одноплатні комп'ютери з попередньо встановленою операційною системою Linux та клієнтським програмним забезпеченням для підключення до сервера кластера по гетерогенних каналах зв'язку (Ethernet, WiFi). На основі даних пристроїв формується віртуальний кластер і координаційний шар кластерів.

### 3.2.2 Шар формування і управління кластером

Шар формування та управління кластером забезпечує реєстрацію, тестування продуктивності і якісних характеристик вузлів, забезпечує підбір пристроїв для формування кластера, а також управління віртуальним кластером за допомогою відповідного модуля. Цей шар знаходиться на окремому пристрої, у нашому випадку даним пристроєм виступає одноплатний комп'ютер Raspberry Pi на базі операційної системи Linux.

Завдяки мікросервісній архітектурі шар представлення результатів реалізований в компоненті «Управління Веб сервісом». Також даний компонент надає дані о поточних зареєстрованих пристроях, призначені їм ролі, характеристики продуктивності і фізичні параметри пристроїв, в тому числі про сформовані віртуальні кластери і виконувані завдання.

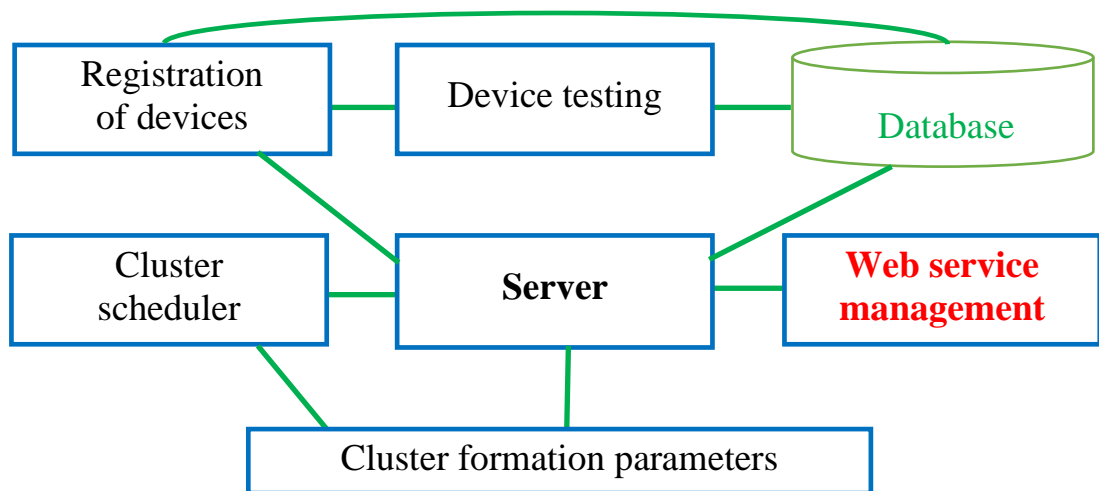


Рисунок 3.4 – Основні компоненти шару формування та управління кластером

База даних на даному шарі зберігає інформацію о підключених пристроях, їх фізичні характеристики, моделі пристроїв, результати тестів, а також призначену роль і місце в системі віртуального кластерів. Додатково до перелічених даних додаються результати роботи віртуальних кластерів, у тому числі призначені завдання, поточний статус вузлів, структуру кластера тощо.

Взаємозв'язки компонент даного шару наведені на рисунок 3.4.

Розглянемо детальніше кожен з компонентів даного шару.

Компонент «Сервер». Даний компонент відповідає за роботу і запуск всіх компонентів системи, їх взаємодію між собою та з операційною системою, в тому числі і за підключення/реєстрацію нових пристроїв. Сервер призначається і налаштовується користувачем вручну.

Компонент «Реєстрація Пристроїв». Всі пристрої перед тим, як бути додані в систему, повинні пройти реєстрацію. Реєстрація пристроїв дозволяє сформувати перелік всіх активних пристроїв, підключених до серверу. Також, даний компонент зберігає інформацію о всіх задіяних пристроях і призначених їм ролях. Перелік зареєстрованих пристроїв зберігається в базі даних, якщо тестування пристрою не проходило, то додається відповідний прапорець.

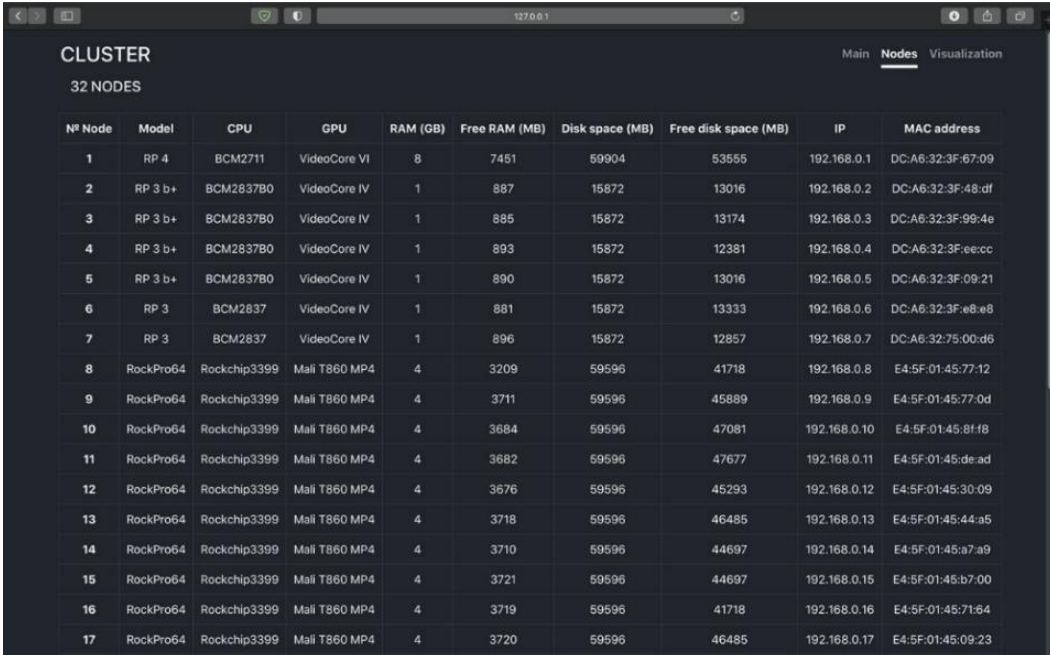
Компонент «Тестування продуктивності пристроїв». Кожний новий пристрій проходить тестування продуктивності на різних обчислювальних задачах, наприклад, таких: тестування швидкості передачі даних по мережі; тестування пристрою кросплатформним програмним забезпеченням в однопоточному і багатопоточному режимах; тестування ОЗП на швидкість читання/запису; тестування на стиск файлу в однопоточному і багатопоточному режимах. За необхідності можна, можливо додати свої тести до вже існуючих. Після проходження даних тестів до кожного пристрою, крім його характеристик, додається запис результатів тестування. Новий сформований список зберігається в базі даних для використання іншими компонентами системи.

Компонент «Планувальник кластера». Компонент служить для завдання користувачем початкових параметрів формування віртуальної групи вузлів (ВГВ), а саме: параметри для координаційного кластера з вказівкою необхідного дискового простору; мінімальну кількість координаційних вузлів; кількість віртуальних кластерів і обчислювальних вузлів в них; перелік модулів для завантаження віртуальних кластерів, що формуються. У

процесі ВГВ будуть формуватися з урахуванням вибору найбільш підходящих по середніх характеристиках вузлів для забезпечення працездатності шару та кластера в цілому.

Компонент «Параметри формування кластера». Даний компонент формує «Координаційний шар кластера» і ВГВ на основі доступних пристроїв та заданих параметрів на основі вибору найбільш підходящих вузлів для кожного шару. Результати попереднього формування віртуального кластера передаються в компонент «Керування веб сервісом» для підтвердження користувачем та/або його редагування шляхом перепризначення вузлів на інші ролі, а при наявності вільних, не призначених вузлів, дозволяє розподілити їх в поточну конфігурацію або зібрати нову, в тому числі призначити роль «дублюючих» вузлів для оперативної заміни вузлів, що відмовили.

Компонент «Управління веб сервісом». Даний компонент системи, що розробляється, має графічний інтерфейс взаємодії з координаційним шаром і віртуальними кластерами. Він дозволяє отримати актуальну інформацію про пристрої, їх характеристики, призначену роль і стан в організації віртуальних кластерів (рисунок 3.5)



CLUSTER									
32 NODES									
Nº Node	Model	CPU	GPU	RAM (GB)	Free RAM (MB)	Disk space (MB)	Free disk space (MB)	IP	MAC address
1	RP 4	BCM2711	VideoCore VI	8	7451	59904	53555	192.168.0.1	DC:A6:32:3F:67:09
2	RP 3 b+	BCM2837B0	VideoCore IV	1	887	15872	13016	192.168.0.2	DC:A6:32:3F:48:df
3	RP 3 b+	BCM2837B0	VideoCore IV	1	885	15872	13174	192.168.0.3	DC:A6:32:3F:99:4e
4	RP 3 b+	BCM2837B0	VideoCore IV	1	893	15872	12381	192.168.0.4	DC:A6:32:3F:ee:cc
5	RP 3 b+	BCM2837B0	VideoCore IV	1	890	15872	13016	192.168.0.5	DC:A6:32:3F:09:21
6	RP 3	BCM2837	VideoCore IV	1	881	15872	13333	192.168.0.6	DC:A6:32:3F:e8:a8
7	RP 3	BCM2837	VideoCore IV	1	896	15872	12857	192.168.0.7	DC:A6:32:75:00:d6
8	RockPro64	Rockchip3399	Mali T860 MP4	4	3209	59596	41718	192.168.0.8	E4:5F:01:45:77:12
9	RockPro64	Rockchip3399	Mali T860 MP4	4	3711	59596	45889	192.168.0.9	E4:5F:01:45:77:0d
10	RockPro64	Rockchip3399	Mali T860 MP4	4	3684	59596	47081	192.168.0.10	E4:5F:01:45:8f:f8
11	RockPro64	Rockchip3399	Mali T860 MP4	4	3682	59596	47677	192.168.0.11	E4:5F:01:45:de:ad
12	RockPro64	Rockchip3399	Mali T860 MP4	4	3676	59596	45293	192.168.0.12	E4:5F:01:45:30:09
13	RockPro64	Rockchip3399	Mali T860 MP4	4	3718	59596	46485	192.168.0.13	E4:5F:01:45:a4:a5
14	RockPro64	Rockchip3399	Mali T860 MP4	4	3710	59596	44697	192.168.0.14	E4:5F:01:45:a7:a9
15	RockPro64	Rockchip3399	Mali T860 MP4	4	3721	59596	44697	192.168.0.15	E4:5F:01:45:b7:00
16	RockPro64	Rockchip3399	Mali T860 MP4	4	3719	59596	41718	192.168.0.16	E4:5F:01:45:71:64
17	RockPro64	Rockchip3399	Mali T860 MP4	4	3720	59596	46485	192.168.0.17	E4:5F:01:45:09:23

Рисунок 3.5 – Відображення зареєстрованих вузлів в веб сервісі

Даний інтерфейс дозволяє поставити вручну параметри формування кластера або, переназначивши ролі для зареєстрованих пристроїв, перемістити їх на іншу позицію, призначивши відповідну роль.

Додатково реалізовано шар візуалізації сформованого віртуального кластера, де центральний вузол А є координаційним вузлом, при цьому помаранчевим кольором відзначені вузли зберігання даних, зелений і темно червоний колір – віртуальні групи кластерів (рисунок 3.6).

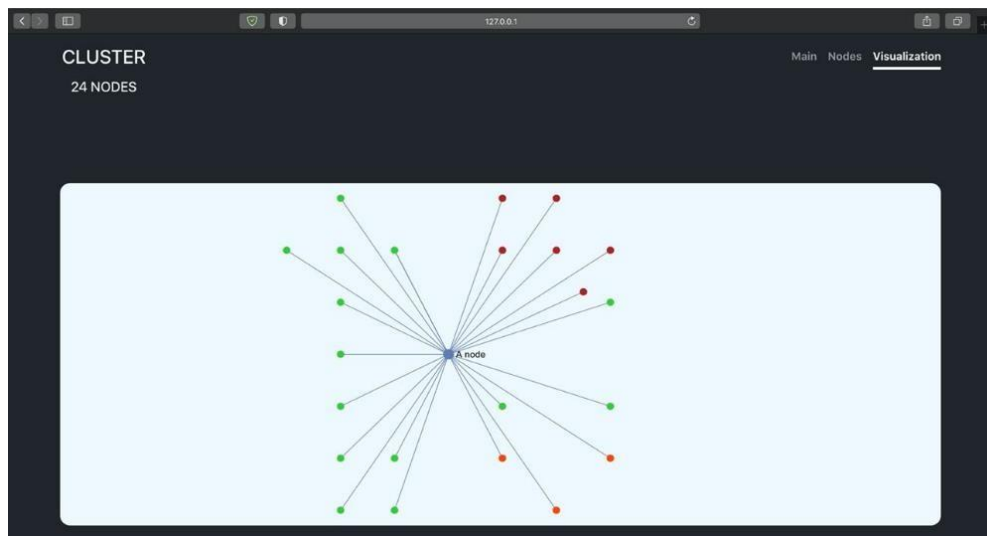


Рисунок 3.6 – Візуалізація сформованого віртуального кластера

### 3.2.3 Координаційний шар кластера

Координаційний шар кластера відповідає за управління віртуальними кластерами, відстеження їх стану, організацію віртуальної пам'яті для зберігання даних, що надходять для обробки, так і за результати обчислень з віртуального кластерів. Цей шар містить один координаційний (керуючий) вузол, що здійснює взаємодію з головними вузлами віртуальних кластерів, а також здійснює контроль за організацією зберігання даних. Координаційний вузол є сполучною ланкою між основним сервером і віртуальними кластерами, виконуючи роль інформаційного агрегатора, що збирає дані про стан віртуального кластера. Інформацію про результати роботи він передає на основний сервер, де користувач через веб інтерфейс може спостерігати за

роботою системи в загалом.

Взаємозв'язки компонент даного шару наведені на рисунок 3.7.

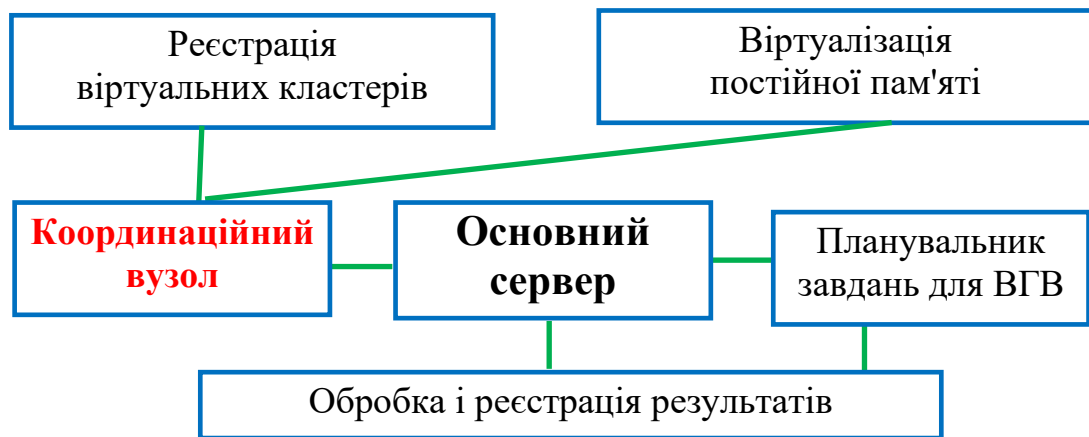


Рисунок 3.7 – Основні компоненти координаційного шару кластеру

Розглянемо детальніше кожен з компонентів даного шару.

Компонент «Сервер». Після призначення пристрою відповідної ролі йому передаються відповідні інструкції для скачування і установки необхідних модулів координаційного вузла. Даний модуль забезпечує працездатність всіх компонентів шару, їх зв'язок і взаємодію з іншими компонентами системи.

Компонент «Реєстрація віртуальних кластерів». У компоненті міститься інформація про всі віртуальні кластери, призначені пристрої та присвоєні ролі, а також виділений ним дисковий простір для зберігання даних. Даний компонент дозволяє отримувати інформацію про статус кожного пристрою та його взаємодії із системою за допомогою логування виконуваних операцій. Інформація про вузли сформованого віртуального кластера дозволяє його переформувати в випадку виходу з ладу вузлів або доповнити обчислювальними вузлами для підвищення продуктивності кластерів.

Компонент «Віртуалізація постійної пам'яті». Оскільки комп'ютери з обмеженими обчислювальними ресурсами мають різноманітні характеристики по низці параметрів, у тому числі й дисковому простору, то

стоїть завдання організувати на базі комп'ютерів з обмеженими ресурсами сформувати сховище даних для забезпечення необхідного простору на основі постійної пам'яті. Для організації зберігання даних вибираються пристрої з максимальним вільним дисковим простором, високою швидкістю передачі даних мережею і мінімальною продуктивністю у порівнянні з іншими пристроями, зареєстрованими в системі (рисунок 3.8).

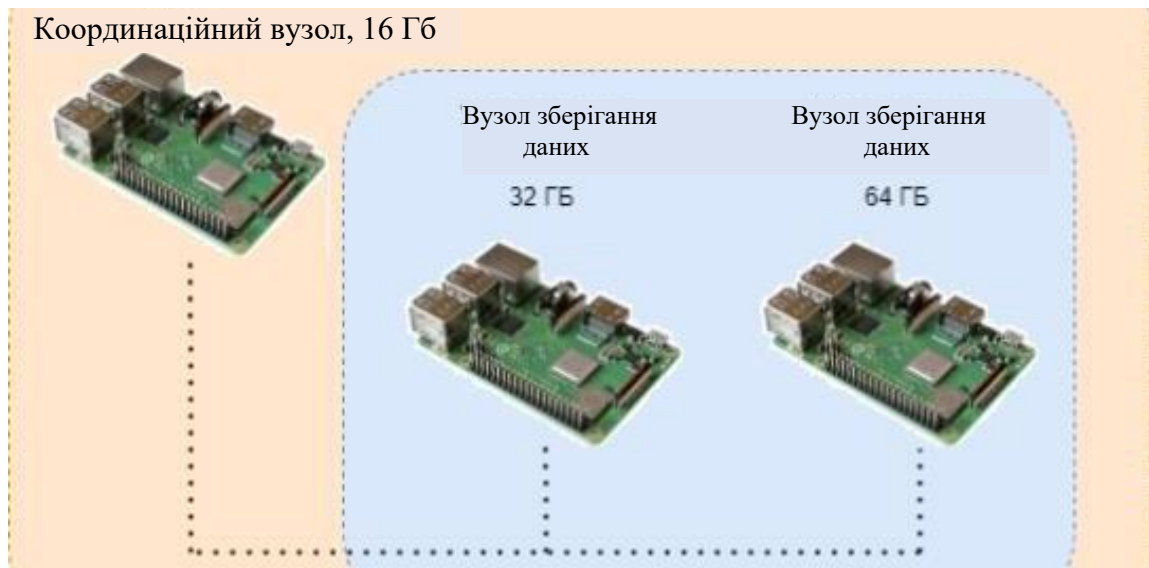


Рисунок 3.8 – Приклад побудови віртуалізації пам'яті

Підключення пристрою з роллю «вузол зберігання даних» до координаційного вузла здійснюється з використанням клієнтської програми SSHFS (Secure SHell FileSystem). Координаційний вузол реєструє виділений дисковий простір вузла зберігання даних та записує на нього необхідні файли, при цьому шлях до файлу зберігається у базі даних на координаційному вузлі. При необхідності можливо збільшити обсяг постійної пам'яті, що надається, шляхом додавання відповідного вузла в координаційний шар з присвоєнням йому відповідної ролі.

Компонент «Планувальник завдань для ВГВ». Планувальник завдань для віртуальної групи вузлів (ВГВ) реєструє призначене користувачем завдання на сформований віртуальний кластер, визначає посилення на вхідні дані, виділений дисковий простір для зберігання даних у компоненті

«Віртуалізація постійної пам'яті». Компонент також відповідає за попередній аналіз даних, що складається з мікросервісів, які відповідають за різні типи даних: зображення/ відеозображення, текстові або дані часових рядів, пов'язаних через API Rest. На цьому шарі на мікросервісах відбувається аналіз типів даних. При невідповідності даних або при виникненні помилок, повідомлення надсилається пристрою, який є джерелом даних, в випадку повторення помилок і неможливості їх усунення відправляється повідомлення користувачеві за допомогою компонента «Управління веб сервісом».

Компонент «Обробка і реєстрація результатів». Даний компонент фактично є «шаром уявлення даних», так як відповідає за виведення результатів залежно від завдання, а також їх реєстрацію та запис на виділений дисковий простір. Інформація о нових результатах передається на шар управління для сповіщення користувача. У тому числі повідомляється про поточний стан вузлів та їх завантаження (процесор, пам'ять, диск, мережа).

### 3.2.4 Шар віртуального кластера

За своєю суттю даний шар архітектури відповідає звичайному кластеру з одноплатних комп'ютерів, котрий може бути зібраний на локальному стенді і складається з головного (керуючого) вузла, підпорядкованих обчислювальних вузлів і комутатора. Різниця складається тільки в тому, що даний шар може включати кілька не пов'язаних між собою віртуальних кластерів, а пристрої, що беруть участь в формуванні кластера, різномірні і взаємодіють в умовах гетерогенної мережі. Кожен віртуальний кластер взаємодіє з координаційним вузлом через головний вузол. Тобто у розробленій архітектурі шар віртуального кластера є частковим компонентом системи на відміну від альтернативних рішень з реалізацією повноцінного кластера на одноплатних комп'ютерах.

Другою відмінністю цього шару є те, що віртуальний кластер може формуватися з гетерогенних пристроїв, подібних по характеристиках і продуктивності в різних задачах. Головний вузол може бути обраний з більш продуктивних пристроїв для забезпечення максимальної стабільності та продуктивності кластера.

Майстер-вузол за допомогою компонента «Сервер» нерозривно пов'язаний з координаційним вузлом, отримуючи від нього дані для аналізу, а також передаючи йому свій поточний стан, якщо майстер-вузол відключиться від мережі, то призначений йому пул завдань буде переданий наступному вузлу, що займе його роль.

Взаємозв'язки компонент даного шару наведені на рисунок 3.9.



Рисунок 3.9 – Основні компоненти шару віртуального кластера

Розглянемо детальніше кожен з компонентів даного шару.

Компонент «Методи і алгоритми». Модуль відповідає за застосовувані методи та алгоритми у виконанні обчислювального завдання. Залежно від того, яка задача, користувач вибирає необхідний модуль, де може задати свої параметри або використовувати попередньо навчені ваги/моделі. Далі модуль завантажується на головний вузол віртуального кластера. Завдяки модульному принципу можна написати та завантажити свій модуль, описавши методи та алгоритми для вирішення поставленого завдання.

Компонент «Планувальник завдань». Даний модуль розподіляє завдання між обчислювальними (робочими) вузлами віртуального кластера. Також планувальник містить інформацію про підключення до серверу

зберігання даних. Усі звернення обчислювальних вузлів до серверу проходять через цей компонент, що дозволяє відстежувати вільні порти. Видача портів відбувається за алгоритмом FIFO (першим прийшов - першим пішов).

Компонент «Робочі процеси». Робочі процеси виконують повний цикл обчислення завдання, а саме виконуючи підключення до планувальника завдань, отримуючи вільний порт підключення до сервера, далі отримуючи дані на обробку з черги завдань та вказівку щодо того, яку задачу потрібно виконати над цими даними. Далі виконується рішення завдання, по завершенню якого процес підключається до планувальника і знову отримує порт для підключення до серверу з метою передачі оброблених даних.

### 3.3 Етапи формування віртуального кластера

Формування віртуального кластера поділено на кілька послідовних етапів. Розглянемо основні методи та алгоритми, що беруть участь у формуванні віртуальної групи пристроїв з самого початку. Першим в роботі кластера задіється шар формування і управління кластером (сервер ВК), він відповідає за:

- реєстрацію і тестування нових пристроїв з присвоєнням пристрою ролі, відповідної характеристикам пристрою та поточним параметрам кластерів;
- підбір параметрів для віртуальних кластерів на основі наявних пристроїв;
- працездатність всіх рівнів віртуальних кластерів;
- взаємодія користувача із сформованим кластером через веб інтерфейс.

Після реєстрації та тестування доступних пристроїв сервер ВК задає початкові параметри для формування ВК, яке починається з координаційного шару. Даний шар складається з координаційного вузла (КВ) та вузлів

зберігання даних. При цьому КВ відповідає за взаємодію сервера та віртуальної групи пристроїв (ВГП), а також за взаємодію з вузлами зберігання даних.

Наступним кроком необхідно призначити майстер вузол, а також підпорядкувати йому придатні обчислювальні вузли, які сформують ВГП. Слід відзначити, що координаційний вузол може обслуговувати кілька ВГП.

При цьому характеристики комп'ютерів можуть істотно відрізнятись, як і інші важливі параметри (швидкість мережі, завантаженість процесора та ОЗУ, кількість вільної постійної пам'яті тощо). Для того, щоб сформувати ВГП більш збалансованою, необхідно підібрати пристрої по подібних параметрах, що відповідає завданню кластеризації.

Основні кроки даних етапів наведені на рисунок 3.10.



Рисунок 3.10 – Основні етапи формування віртуального кластера

Розглянемо детальніше деякі суттєві складові даного процесу, такі як реєстрація, тестування та призначення ролі пристрою.

Після запуску сервера і завантаження необхідних компонентів

керуючий вузол починає приймати зовнішні запити на реєстрацію пристроїв із встановленою клієнтською частиною. Усі підключені пристрої проходять реєстрацію і відправляють основні дані пристрої (модель пристрою, mac адреса, об'єм постійної пам'яті тощо), після чого одержують завдання на проведення тестів для визначення продуктивності.

Після отримання списку пристроїв і результатів тестування починається формування кластера, основні бажані параметри для віртуальної групи пристроїв (ВГП) (необхідна кількість пам'яті, кількість ВГП, кількість обчислювальних вузлів в ній тощо) користувач може поставити заздалегідь або ґрунтуючись на отриманих даних про зареєстровані пристрої через веб інтерфейс. На даному етапі всім задіяним пристроям присвоюються їхня роль у ВГП, займане становище в кожному шарі і передаються необхідні модулі для завантаження.

Після завантаження всіх необхідних модулів (компонентів) вузол відправляє сигнал про готовність на сервер. Якщо вузол отримав роль «Сервер», він чекатиме підключення вузлів до повного формування шару згідно з отриманими даними. Якщо пристрій (вузол) має іншу роль, то по завершенні завантажень і запуску всіх модулів (компонентів), необхідних для здійснення свого завдання, він відправляє сигнал о готовності на сервер і чекає від нього даних щодо підключення до потрібного вузла.

Процес реєстрації пристрою складається з кількох етапів і доступний для пристроїв з встановленим клієнтським програмним забезпеченням. Після запуску і появи в мережі сервера клієнт підключається до нього і передає дані про себе в вигляді IP і MAC адрес для однозначної ідентифікації даного пристрою в майбутньому. Після реєстрації і занесення відповідних даних в базу даних компонент «Реєстрація пристроїв» передає команду в компонент «Тестування» про пристрої для призначення відповідних модулів для проведення тестування продуктивності та відправляє її на сервер, де вже в порядку черги вона передається на відповідний пристрій. Результати тестування передаються в компонент «Тестування», звідки дані вирушають

до реєстратору з занесенням їх в базу даних. Зареєстрований пристрій додається до списку, що бере участь в формуванні кластера.

У розробленій архітектурі виділяються два шари: шар координації кластера і шар ВГП. На кожен з цих шарів призначається пристрій з певною роллю.

Для шару координації кластера призначається координаційний вузол - пристрій з цією роллю відповідає за розподіл завдань між ВГП, а також за організацію віртуалізації зберігання даних і розподіл файлів між вузлами зберігання даних. У тому числі здійснюється реєстрація і зберігання результатів обробки даних. Вузол зберігання даних - пристрій, що призначається для організації постійного дискового простору для зберігання даних.

Шар віртуальної групи пристроїв має головний вузол, він же «Майстер Вузол» (МВ), напряму пов'язаний з координаційним Вузлом, керує станом кластера, розподіляє завдання між обчислювальними вузлами. На цьому шарі також призначаються віртуальні обчислювальні пристрої (ВОП) – пристрої, призначені на цю роль, виконують завдання та результати повертають МВ.

### 3.4 Алгоритми і методи формування віртуального кластера

Алгоритм балансування розподілу завдань для вузлів зберігання даних. З метою збільшення швидкості передачі даних від вузлів зберігання даних до обчислювальних вузлів в рамках обмежених ресурсів одноплатних комп'ютерів був розроблений евристичний алгоритм розподілу даних по вузлах в залежності від кількості даних та швидкості мережевої взаємодії пристрою.

Крок 1. Визначення кількості вузлів зберігання даних в залежності від обсягу вихідних даних до кількості поточних зареєстрованих пристроїв.

Крок 2. Реєстрація підходящих вузлів зберігання даних.

Крок 3. Розрахунок загальної швидкості передачі даних і відсотку від

суми для кожного пристрою.

Крок 4. Розподіл файлів, що передаються на вузол зберігання.

Крок 5. Передача даних на відповідні вузли.

Крок 6. Створення черги завдань з рівномірним розподілом навантаження на вузли зберігання даних.

Алгоритм заміни головного вузла. Майстер-вузол може відмовити та втратити з'єднання з кластером, тому необхідно відстежувати його стан для мінімізації втрат продуктивності ВДУ, пов'язаного з його простоем. Якщо обчислювальний вузол не може отримати доступ до Майстер-вузла, він надсилає дані о події координаційному вузлу. Координаційний вузол на основі отриманих даних перевіряє правильність параметрів підключення обчислювального вузла і якщо вони вірні, то здійснює перевірку доступності Майстер-вузла. При підтвердженні непрацездатності поточного Майстер-вузла ініціюється алгоритм його заміни на найближчий вузол із найбільш наближеного до характеристикам до заміненого Майстер-вузлу. Новому вузлу передаються всі необхідні модулі для завантаження, а також черга завдань на виконання обчислень.

Після визначення ключових вузлів та призначення ним відповідної ролі у роботі системи зі списку незадіяних пристроїв необхідно сформулювати ВГП. Для вирішення проблеми підбору пристроїв за характеристиками та продуктивністю були використані алгоритми кластеризації даних без вчителя, основною відмінністю яких є відсутність розмітки для даних, а також неочевидність вимірювання якості методів через відсутність прямолінійних метрик, застосовуваних у задачах з вчителем.

Метод K- Means. Цей метод є одним самих простих і поширених методів кластеризації. Він мінімізує сумарне квадратичне відхилення точок кластерів від центрів цих кластерів. Основною проблемою даного методу є те, що кількість кластерів потрібно знати заздалегідь. При цьому результат буде залежати від вихідних центрів кластерів, але їх оптимальний вибір не відомий. У випадку рішення завдання кластеризації достатньо складно

підібрати критерій оптимізації, проте при цьому необхідно вибрати кількість кластерів, випробувавши різні комбінації. Це дозволяє розмістити точки навколо центрів своїх кластерів. Але необхідно використовувати евристичний метод для визначення кількості кластерів.

Метод агломеративної кластеризації. Відмінною рисою даного методу від попереднього є відсутність вимог до визначенню числа кластерів. Алгоритм методу описується такими кроками.

Крок 1. Формуємо свій кластер на кожну точку.

Крок 2. Виконуємо сортування попарних відстаней елементів між центрами кластерів за зростанням.

Крок 3. Об'єднуємо найближчі кластери в один. Вираховуємо новий центр кластерів.

Крок 4. Циклічно повторюємо кроки 2 і 3 до виконання умови закінчення.

Метод спектральної кластеризації. Першим кроком алгоритму методу спектральної кластеризації є визначення матриці подібних елементів по відношенню «схожість» елементів. Дана матриця описує повний граф з вершинами елементів і ребрами, що зв'язує їх, з вказівкою ваги їх «схожість»: чим коротше ребро, тим більше схожі ознаки. Розглянемо алгоритм.

Крок 1. Обчислення нормалізованого Лапласіана.

Крок 2. Розрахунок перших  $k$  власних векторів.

Крок 3. Формуємо матрицю, створену першими  $k$  власними векторами.

Крок 4. Кластеризація вузлів графа на основі цієї матриці.

Метод головних компонент. Даний метод використовується у разі необхідності зменшити розмірність даних, втративши найменшу кількість інформації. Для того щоб знизити розмірність з  $n$  в  $k$ ,  $k < n$ , необхідно вибрати найкращі  $k$  осей, відсортованих по спаданню дисперсії вздовж осей.

Першим кроком необхідно розрахувати дисперсію та коваріацію вихідних елементів, використовуючи матрицю коваріації. Подальшим кроком розкладаємо матрицю коваріації як добуток прямої та транспонованої

матриць. З відношення Релея визначаємо, що досягнення максимальної варіації проходить вздовж власного вектора матриці, який відповідає максимальному власному значенню. Отже, обираємо  $k$  власних векторів матриці, які і будуть головними компонентами. Для отримання проекції даних в ортогональному базисі компонентів необхідно помножити матрицю даних на ці компоненти. Транспонуємо матрицю даних та матрицю векторів головних компонент. При цьому, якщо в результаті перетворень кількість компонент було менше розмірності вихідного простору, то відбувається втрата частини інформації, але завдяки цьому методу втрачається мінімальна кількість інформації.

Для оцінки результатів кластеризації були розглянуті наступні метрики ефективності:

*Silhouette Coefficient* – метрика визначається для кожної вибірки і складається з двох оцінок, причому більш високий коефіцієнт відноситься до моделі з краще визначеними даними.

*Calinski-Harabasz Index* – метрика, відома як критерій відношення відхилень, представляє собою відношення суми дисперсії між кластерами і дисперсії всередині кластерів для всіх кластерів. Результат кластеризації тим краще, чим більше значення критерію.

*Davies-Bouldin Index* – метрика, що визначає середню «схожість» між кластерами, де подібність – це міра, яка порівнює відстань між кластерами з розміром самих кластерів. Нижчий індекс Девіса-Болдіна відноситься до моделі з кращим поділом кластерів.

При формуванні обчислювального шару віртуальних груп вузлів враховується гетерогенність пристроїв і їх підбір по приблизно однаковим характеристикам і продуктивність. Розглянемо кілька методів розподілу вузлів.

Закон Амдала. Прискорення паралельних обчислень щодо послідовних може бути знайдено по зваженому закону Амдала. Для гетерогенних обчислювальних вузлів він враховує продуктивність та пропускну здатність

мережі. Тоді найкраща конфігурація ВДУ є та, в якій продуктивність кластера з мінімальним приростом продуктивності максимальна. Це завдання у загальному вигляді має величезну обчислювальну складність, оскільки має на увазі перебір всіх підмножин множини вузлів. Для скорочення простору перебору можна, можливо скористатися евристиками.

Емпіричне заповнення. Цей підхід передбачає заповнення віртуального кластера обчислювальними вузлами доти, доки виконується умова обмеження приросту прискорення обчислення. Обмеженням може виступати приріст швидкості виконання тестового завдання, наприклад припинити додавання обчислювальних вузлів до групи, якщо при додаванні вузла тестове завдання було вирішено менше, ніж з 5% прискоренням в порівнянні з попереднім результатом. Таким чином, можна налаштувати «оптимальну» продуктивність віртуального кластера. для заздалегідь взятого завдання з гетерогенними пристроями. Мінусом даного методу є необхідність кожного разу підбирати вузли віртуального кластера для нового завдання.

Мета-навчання. Підхід з «мета-навчанням» заснований на заздалегідь відомих комбінаціях пристроїв та необхідних параметрах для «збору віртуального кластера», тобто підбір віртуальних обчислювальних вузлів буде заснований на «шаблонах» протестованих пристроїв у складі кластера для заздалегідь певного алгоритму. При накопиченні великої кількості тестових даних даний метод дозволить формувати віртуальний кластер з існуючих пристроїв, орієнтуючись на найбільш оптимальні шаблони, що дозволить достатньо швидко збирати ефективні віртуальні кластери під поточні завдання та також швидко перезбирати під нові. Істотним недоліком даного підходу є необхідність провести різнобічні тести на великій кількості комбінацій пристроїв, а при додаванні нової моделі пристрої, суттєво різною від вже тестованих, буде потрібно провести нові випробування.

Заповнення вузлів з застосуванням підходу кластеризації вузлів. У рамках дослідження передбачається використання вже задіяних в роботі

вузлів в мережі Інтернету Речей, отже дані пристрої вже виконують якісь завдання на постійній основі і необхідно також враховувати їх вільні ресурси, також вузли різні за своїми характеристиками.

У цьому підході заповнення віртуальних груп вузлів засноване на розподілі вузлів у групи за схожими характеристиками і ресурсами доступних вузлів, що дозволить сформувати групи під певні розв'язувані завдання. Заповнення віртуальних груп відбувається поетапно, для кожної групи окремо, і використовується загальний перелік вузлів.

### 3.5 Порівняння віртуального і класичного кластерів

Раніше розглядалися класичні архітектури побудови кластерних систем на основі комп'ютерів з обмеженими ресурсами, відмінною особливістю яких є локалізація доступних ресурсів, гомогенність мережних з'єднань, централізація зберігання даних. Проте, враховуючи специфіку комп'ютерів з обмеженими ресурсами, дані архітектури не підходять для організації віртуального кластера, тому окремі компоненти системи були переглянуті і реалізовані з урахуванням неоднорідності доступних ресурсів та гетерогенності мережі.

Зберігання даних у сформованому віртуальному кластері організується на окремому шарі архітектури, за організацію доступної пам'яті відповідає координаційний вузол (КУ), також він створює чергу завдань та організує передачу даних від вузлів зберігання до віртуальних обчислювальних пристроїв та назад. У класичному кластері за це відповідає майстер-вузол (МУ) з заздалегідь заданим розділом постійної пам'яті.

За розподіл завдань на ВГП відповідає також координаційний вузол, котрий створює чергу завдань і пов'язує певний вузол зберігання з ВГП для обміну даними. Виконані завдання видаляються з черги, таким чином, якщо ВГП через збій не виконав завдання або повідомив про помилку, то завдання буде відправлено повторно. У класичному кластері за розподіл та

відстеження виконання завдань відповідає майстер-вузол, якщо у кластері присутні координаційні вузли, то вони розподіляють завдання на обчислювальні вузли.

Додавання пристроїв в кластер відбувається з зареєстрованих, протестованих, доступних вузлів на основі їх розподілу по доступних ресурсах та призначеним ролям, відповідно визначеним у системі. Кожен головний вузол відповідає за формування шару відокремлено від інших. Координаційний вузол організує область зберігання даних з списку вузлів зберігання, розподіляючи дані на них, при необхідності запитуючи додаткові вузли. ВГП формується з запитом майстер-вузлів до координаційного вузлу для визначення вільних та відповідних обчислювальних вузлів. При цьому в класичному кластері кількість пристроїв відома заздалегідь, параметри формування кластера задаються вручну.

Розподіл вузлів ВГП відрізняється від класичного кластера тим, що формується в залежності від поставленою завдання і на основі схожих характеристик вузлів для досягнення. При цьому кількість ВГП визначається на основі мета-навчання, вручну або на основі доступних вузлів.

### 3.6 Тестування віртуального кластера

Для тестування був сформований віртуальний кластер для 5000 пристроїв, що включають 12 різних моделей одноплатних комп'ютерів, при цьому дані тестів пристроїв отримано з відкритих джерел. У координаційному шарі розглядалися дві ролі: координаційний вузол та вузол зберігання даних. У першу чергу був виділений координаційний вузол і вже до нього призначені вузли зберігання даних.

При проведенні підбору пристроїв для ролі обчислювальних вузлів на згенерованих даних якісна характеристика пристроїв генерувалася на таких параметрах: продуктивність в однопоточному і багатопоточному тестах, вільна оперативна пам'ять, швидкість мережевої взаємодії. Через порівняння

різних величин була використана нормалізація. Завдання нормалізації було зведено до наступної формули:

$$X_{норм} = \frac{X_i - X_{зсув}}{\Delta X}, \quad (3.1)$$

де  $X_i$  – поточне значення;  $X_{зсув}$  – величина зсуву значень;  $\Delta X$  – величина інтервалу, котрий буде перетворений до одиничного значення, це різниця максимального та мінімального значень нормалізуемого параметра.

Візуальне подання за даними критеріями представлено на рисунок 3.11 з використанням тестів в однопоточному режимі і рисунок 3.12 в багатопотоковому режимі. Зниження розмірності з метою зменшення числа змінних для найкращого представлення даних було здійснено за методом головних компонент (рисунок 3.13).

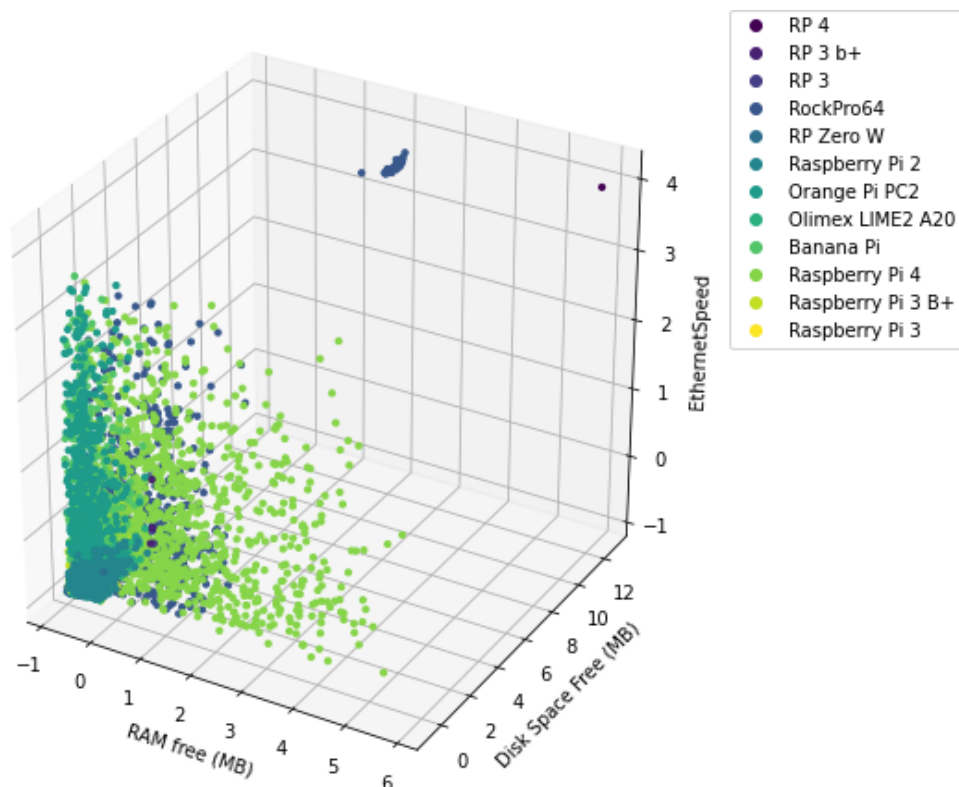


Рисунок 3.11 – Візуалізація за 3 критеріями для обчислювальних вузлів в однопотоковому тесті

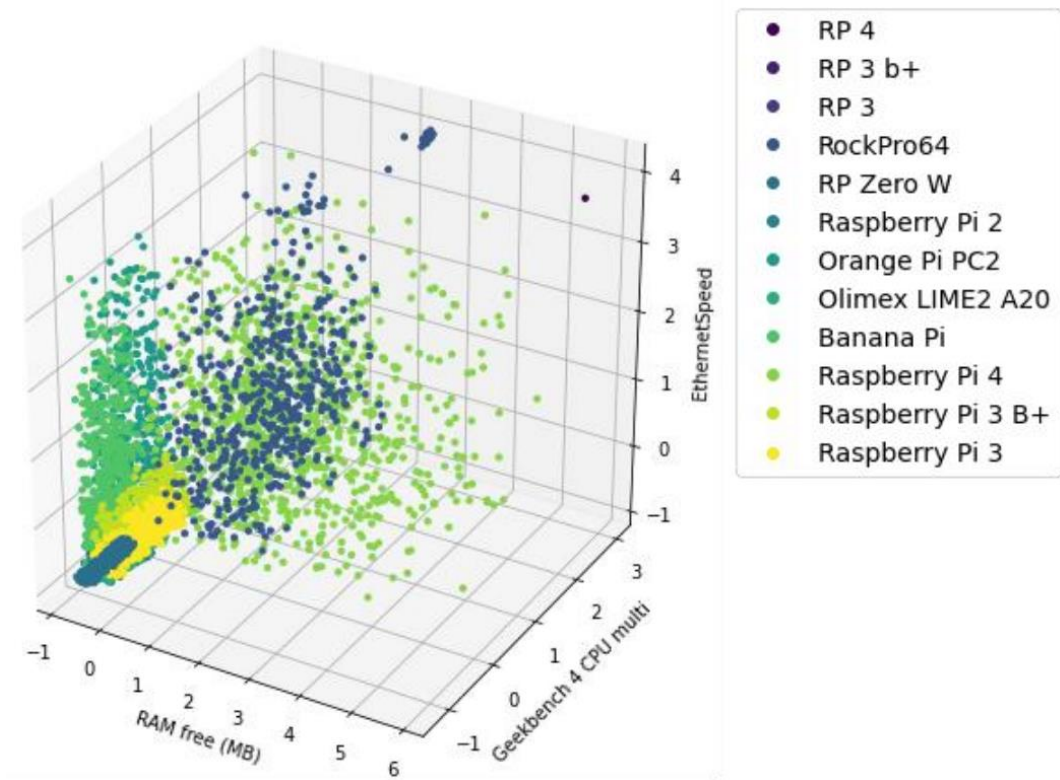


Рисунок 3.12 – Візуалізація за 3 критеріями  
для обчислювальних вузлів в багатопотоковому тесті

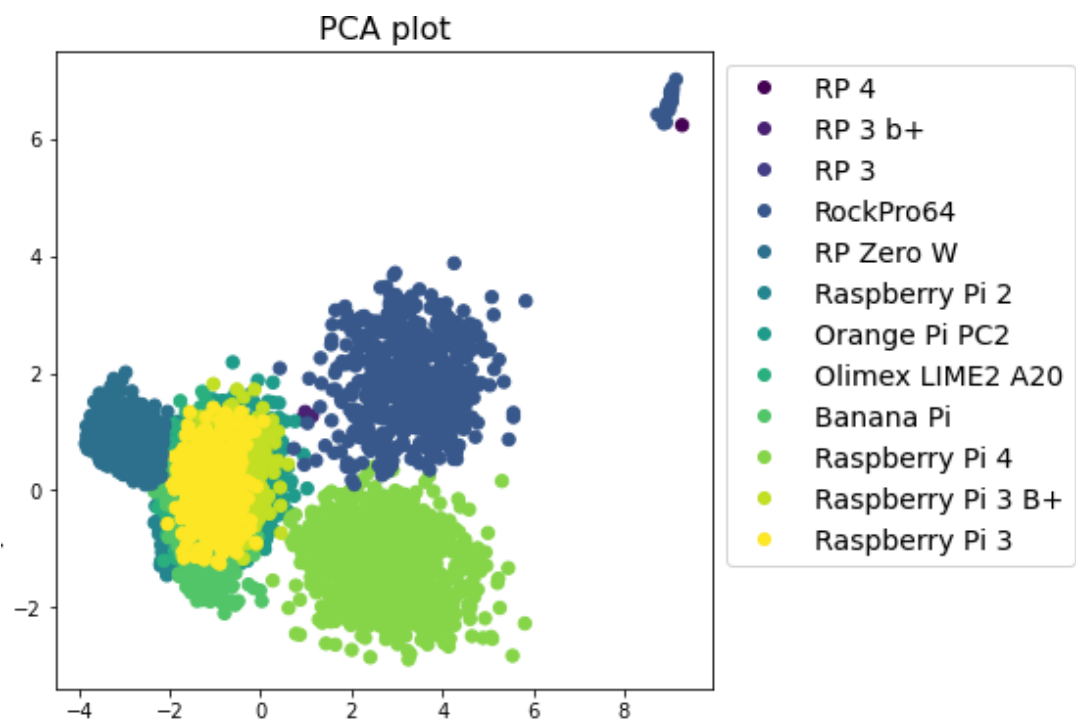


Рисунок 3.13 – Дані в просторі головних компонент

З графіка рисунок 3.13 видно, що деякі моделі пристроїв схожі між собою і на графіку вони виявилися розташованими приблизно в одній

області. При цьому є границі перетинів, які демонструють схожість елементів між собою. Тому необхідно застосувати алгоритм кластеризації для отриманої множини комп'ютерів з обмеженими обчислювальними ресурсами. Для порівняння розглянемо 3 алгоритми кластеризації: k-means, агломеративна кластеризація, спектральна кластеризація.

K-means, перед застосуванням даного алгоритму необхідно визначити кількість кластерів. Вибір кількості кластерів за допомогою інерційного методу представлений на рисунок 3.14, з якого можемо зробити, що найкраще число кластерів складає 5.

Результати k-means в вихідному просторі з візуалізацією у просторі (ОЗУ, CPU однопоточному тісті, мережевою швидкістю) представлені на рисунок 3.15. Результати k-means в просторі головних компонент наочніше демонструють кластеризацію елементів і об'єднують усі пристрої по приблизно однакових характеристиках в 5 кластери (рисунок 3.16).

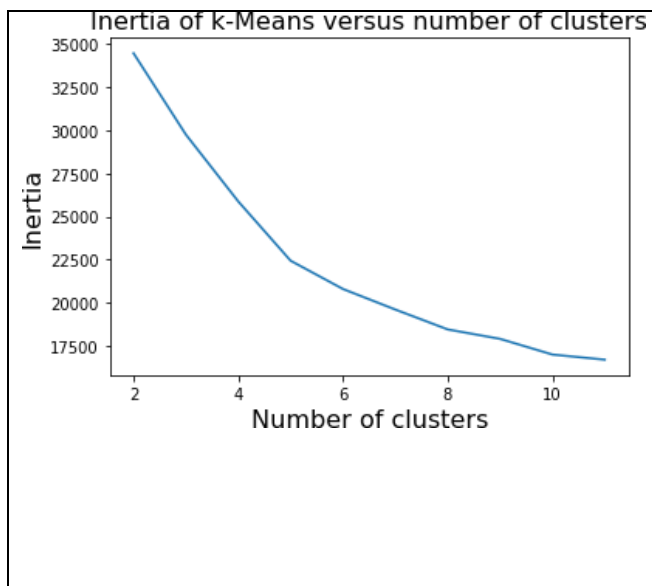


Рисунок 3.14 – Інерція в залежності від числа кластерів

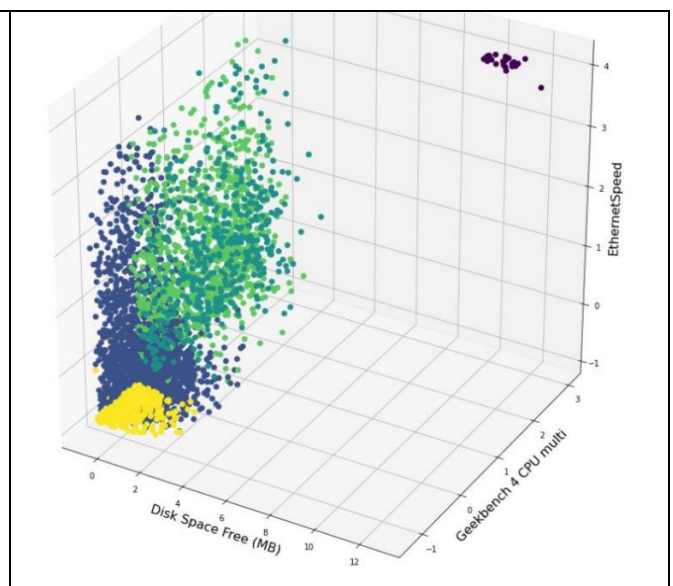


Рисунок 3.15 – Результати k-means у вхідному просторі даних

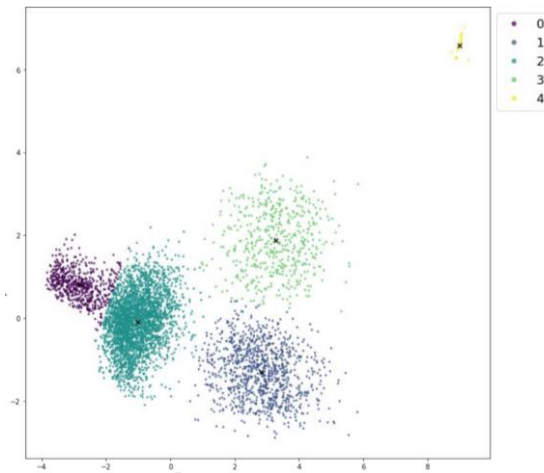


Рисунок 3.16 – Результати k-means у просторі головних компонентів

Наступним розглянемо алгоритм агломеративної кластеризації. Так як нормалізацію вихідних даних була виконано на попередньому кроку, то збудуємо візуалізацію у вхідному просторі даних (рисунок 3.17). Також агломеративна кластеризація в просторі головних компонент демонструє подібну кластеризацію з попереднім алгоритмом і виділяє 5 кластерів (рисунок 3. 18).

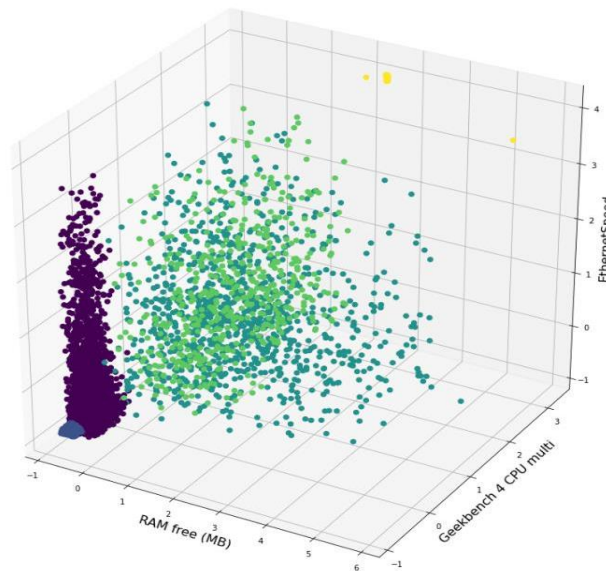


Рисунок 3.17 – Агломеративна кластеризація у вхідному просторі даних

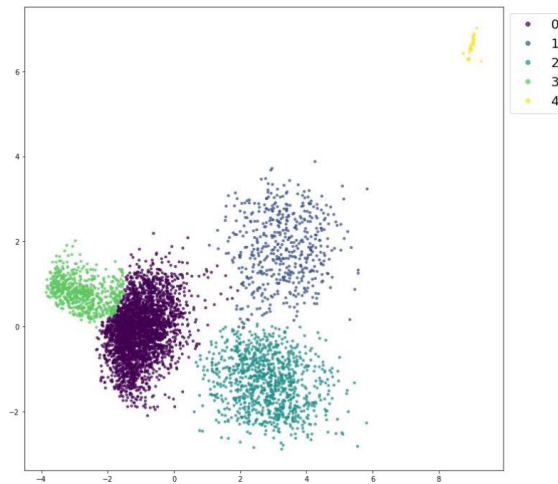


Рисунок 3.18 – Агломеративна кластеризація у просторі головних компонент

Останній алгоритм – це спектральна кластеризація. Побудуємо візуалізацію у вихідному просторі даних (рисунок 3.19). Також спектральна кластеризація в просторі головних компонент демонструє схожу кластеризацію з попередніми алгоритмами (рисунок 3.20).

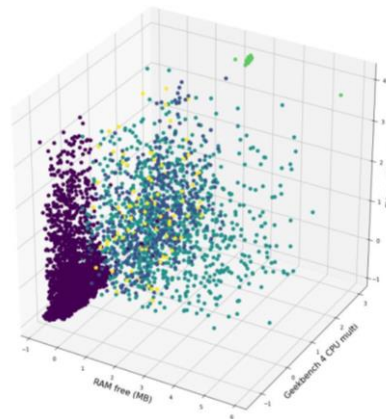


Рисунок 3.19 – Спектральна кластеризація у вхідному просторі даних

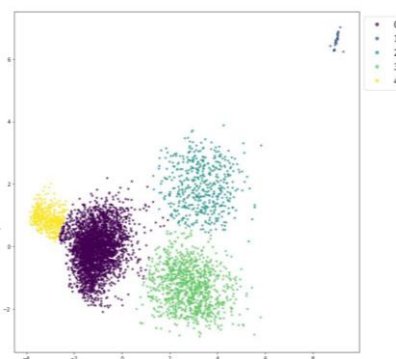


Рисунок 3.20 – Спектральна кластеризація у просторі головних компонент

Після побудови кластеризації необхідно оцінити її результати метриками, описаними вище для кластеризації у вхідному просторі (таблиця 3.1) і в просторі головних компонент (таблиця 3.2).

Таблиця 3.1 – Результати метрик у вхідному просторі

	Silhouette	Davies-Bouldin	Calinski&Harabasz
K- Means	0.404	1.093	1844.8
Agglomerative	0.413	1.085	1835.3
Spectral	0.406	1.920	1404.8

Таблиця 3.2 – Результати метрик в просторі головних компонент

	Silhouette	Davies-Bouldin	Calinski&Harabasz
K- Means	0.618	0.556	8148.6
Agglomerative	0.604	0.574	7998.5
Spectral	0.605	0.602	7742.5

Варто зазначити, що метод основних компонентів відпрацював добре. Оскільки скупчення точок виявилось достатньо далеко друг від друга і відповідно усі три методи кластеризації побудували схожі кластери. Це пояснює і схожі результати метрик.

Найбільш добре відпрацювали методи k-Means і агломеративної кластеризації, які показали схожі результати і відрізняються на рівні похибки. При цьому алгоритм спектральної кластеризації показав найгірший результат у метриках Davies-Bouldin і Calinski&Harabasz.

Оскільки алгоритми кластеризації відпрацювали досить добре, а різниця метрик знаходиться на рівні похибки, то вибір якогось одного алгоритму кластеризації стоїть виробляти по швидкості його роботи. Час виконання алгоритмів кластеризації представлено в таблиці 3.3, результати якої демонструють явну перевагу алгоритму k-Means над двома іншими.

Таблиця 3.3 – Час виконання алгоритмів кластеризації

Назва алгоритму	Час виконання (с)
K- Means	0,66
Agglomerative Clustering	1,39
Spectral Clustering	6,13

Після визначення всіх доступних пристроїв та призначення їм відповідної ролі можна їх використовувати для побудови віртуального кластеру. Від самого початку визначаються координаційні вузли і майстер-вузли, далі йде заповнення координаційного шару вузлами зберігання до досягнення заздалегідь заданих параметрів. Обчислювальні вузли заповнюють віртуальну групу вузлів згідно заздалегідь проведеної кластеризації.

### 3.7 Порівняльне тестування віртуального та класичного кластерів

Перше тестування віртуального кластера проводилося на сформованому вручну віртуальному кластері для невеликої кількості комп'ютерів з обмеженими обчислювальними ресурсами.

Віртуальний кластер мав таку структуру:

- координаційний вузол;
- перша віртуальна група вузлів складається з 12 вузлів, включаючи майстер-вузол;
- друга віртуальна група вузлів складається з 12 вузлів, включаючи майстер-вузол.

Другий кластер був побудований в класичному виконанні і складається з 25 вузлів, включаючи майстер-вузол.

Тестування продуктивності проводилося на класичній задачі додавання матриць розміром: 100, 1000, 10000, 100000. Матриці були згенеровані один

раз випадковим чином.

Для сформованого віртуального кластера координаційний вузол виступає першим балансувальником завдань і доступні завдання віддає майстер-вузлам, які розподіляють їх між вільними вузлами.

У класичному кластері за зберігання даних та розподіл завдань відповідає майстер-вузол. Результати тестування представлені на рисунок 3.21.

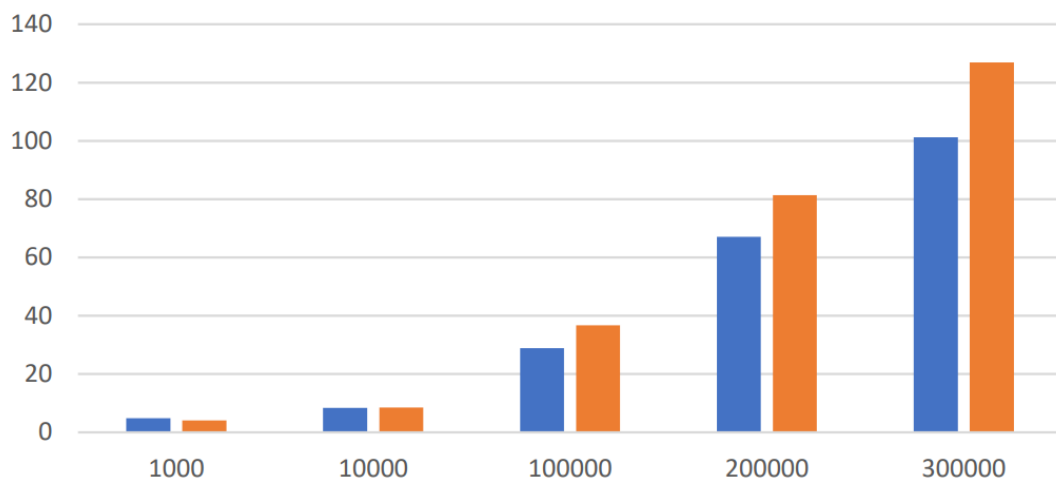


Рисунок 3.21 – Результати тестування кластерів у тесті додавання матриць:

синій колір – час виконання завдань віртуальним кластером;

помаранчевий колір – час виконання завдань класичним кластером

Друге тестування віртуального кластера проводилося з використанням методу емпіричного заповнення обчислювальними вузлами шару віртуальних груп вузлів та перебором усіх доступних варіантів для 25 різних пристроїв, що також дозволило закласти базу для методу «мета-навчання». Тестування проводилося на розпізнаванні зображень з використанням згорткової нейронної мережі з навченої моделі. У ході проведення тестування підбору різних конфігурацій кластера було поставлено умова, що на кожен майстер-вузол повинно доводитися мінімум два обчислювальних вузла, отже мінімальна ВГВ складається з трьох пристроїв, а для 25 доступних для тестування пристроїв максимальна кількість віртуальних

кластерів складає 8. При цьому тестування починається з 1 віртуального кластера з 3 пристроїв, після проходження тесту додається новий вільний пристрій. Було введено обмеження: процес діє поки приріст продуктивності в тесті не буде менше 5% від додавання нового пристрою у віртуальний кластер. Як тільки з'являється таке обмеження, із вільних пристроїв (якщо вони є), створюється новий віртуальний кластер в доповнення до існуючого, але в поточному тесті будуть задіяні усі доступні пристрої.

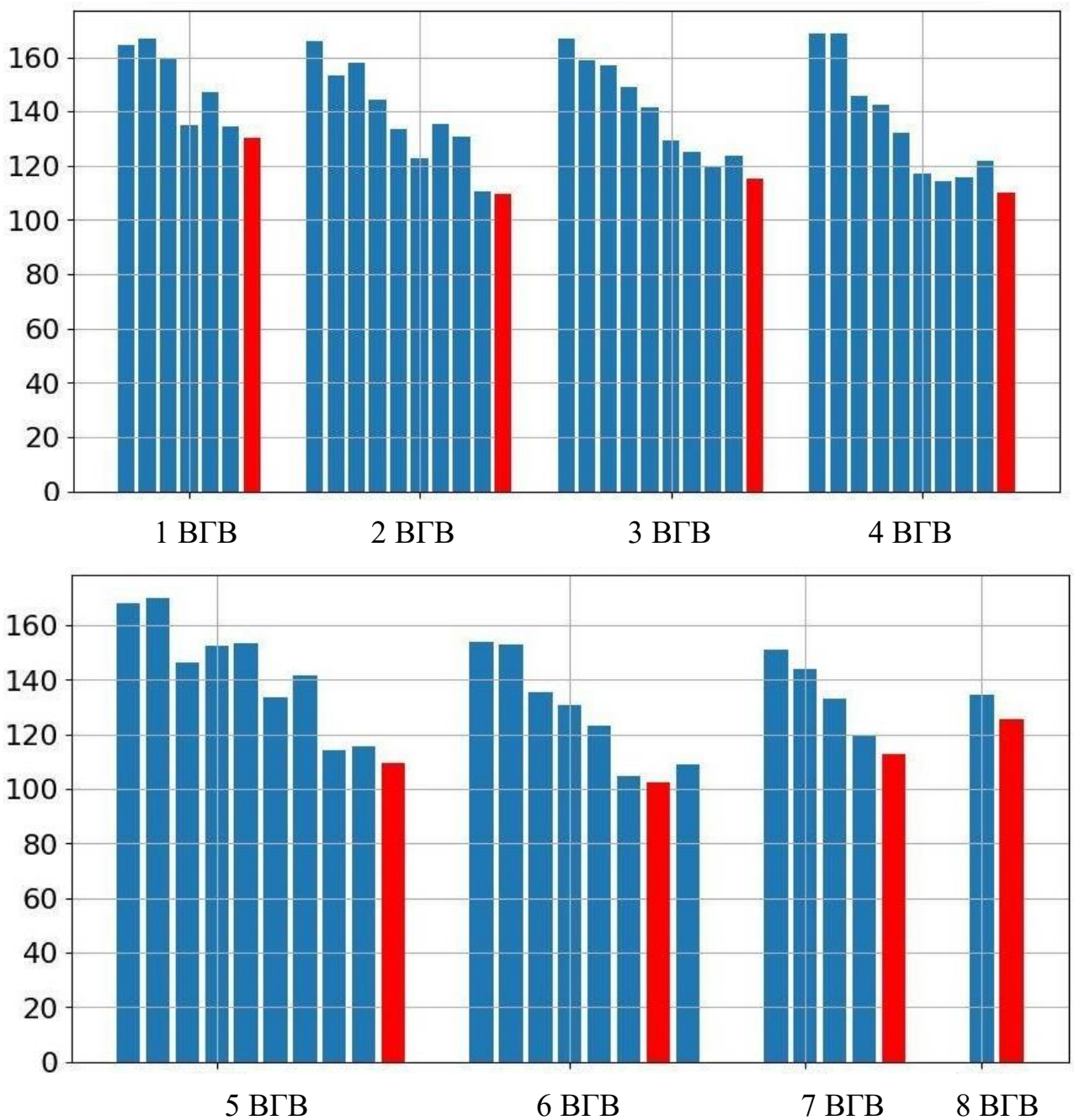


Рисунок 3.22 – Результати тестування різних конфігурацій кластера

Якщо після проходження тесту вільні пристрої закінчилися, але не було досягнуто максимальне обмеження за кількістю кластерів, то починається новий тест, але вже з самого початку створеними двома кластерами і т. д. до досягнення обмеження.

Результати тестування представлені на рисунку 3.22, де по осі X вказана кількість ВГВ з різними конфігураціями вузлів (випадок 1 ВГВ відповідає класичному кластеру), а по осі Y – час виконання тесту на даній конфігурації. Звісно, результат виконання тим краще, чим менше час виконання тіста. Червоним кольором виділена найкраща комбінація серед однотипних конфігурацій.

Найкращі результати тестування, вибрані серед однотипних конфігурацій ВГВ, представлені на рисунку 3.23.

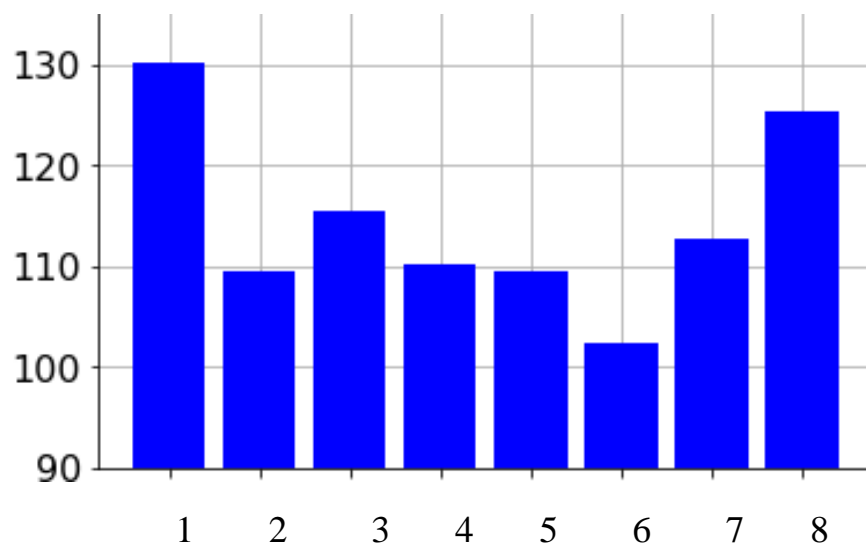


Рисунок 3.23 – Найкращі результати тестування в різних конфігураціях ВГВ

Найкращим є віртуальний кластер з конфігурацією, що складається з 6 ВГВ та 24 пристроїв, в кожен ВГВ входить 4 пристрої відповідно.

## ВИСНОВКИ

Сукупність отриманих у роботі результатів дозволило вирішити актуальне науково-технічне завдання удосконалення методу віртуалізації граничного шару одноплатних комп'ютерів Інтернету речей.

В результаті проведених досліджень отримані такі наукові та практичні результати.

1 Проведений аналіз застосування комп'ютерів з обмеженими обчислювальними можливостями у мережах Інтернету речей. У ході проведеного аналізу були розглянуті одноплатні комп'ютери та можливості їхнього застосування у системах Інтернету речей. Проведено огляд існуючих одноплатних комп'ютерів і їх застосування в структурі кластерів, що беруть участь в освітній сфері, а також у граничних обчисленнях систем Інтернету речей, у тому числі відзначений можливий підхід до створення компактних кластерних модульних систем на одній зв'язуючій платі. Існуючі на даний момент рішення засновані на експериментальних рішеннях організації кластерів в рамках однієї мережі і використанні однієї моделі одноплатного комп'ютера. На основі проведеного аналізу зроблено висновок, що поставлене в роботі завдання щодо розроблення методу організації віртуального кластера, який формується на основі одноплатних комп'ютерів, є актуальним, а його вирішення дозволить більш ефективно використовувати ресурси одноплатних комп'ютерів для організації граничних обчислень.

2 Обґрунтований вибір методів і алгоритмів побудови віртуального кластеру. Розглянуто основні методи пересилання повідомлень між пристроями. Основна проблема пов'язана з вибором відповідного методу передачі даних в залежності від розв'язуваного завдання. Розглянуто популярні алгоритми балансування навантаження. При виборі певного алгоритму слід врахувати особливості його застосування для координуючих вузлів, що виконують певні підзавдання. Розглянуто концепцію

використання вільних ресурсів одноплатних комп'ютерів вже існуючої інфраструктури.

3 Сформована архітектура віртуального кластера комп'ютерів з обмеженими ресурсами. У розробленій архітектурі використовуються чотири основних шари, при цьому шар пристроїв містить зареєстровані в системі пристрої. Інші три шари спроектовані з урахуванням застосування елементів модульної та мікросервісної архітектури. Для кожного шару визначено і описані вузли, а також описані пред'явлені до них вимоги за ключовими критеріями.

4 Удосконалений метод віртуалізації граничного шару одноплатних комп'ютерів Інтернету речей за рахунок побудови віртуального кластеру гетерогенних пристроїв з обмеженими обчислювальними ресурсами, що надало вигоду за часом виконання завдань в середньому на 8%. Розроблено алгоритм формування віртуального кластера для ключових вузлів системи, а саме для координаційного вузла, вузла зберігання даних та майстер-вузла. Для кожного типу вузлів розглянуто ключові критерії. Запропоновано метод кластеризації для вибору обчислювальних вузлів з всього різноманіття пристроїв. Також розглянуті якісні метрики для оцінки кластеризації. Продемонстровано спроможність рішення використання віртуального кластера, а також продемонстровано ефективність обчислень на прикладі розробленого сценарію експериментальної оцінки продуктивності розподіленого віртуального кластера в умовах гетерогенної мережі в завданнях додавання матриць великої розмірності та в задачі розпізнавання зображень з використанням доступних ресурсів. Проведено експериментальну перевірку розроблених методів і алгоритмів формування віртуального кластера в різних конфігураціях, використовуючи обчислювальні завдання для оцінки продуктивності.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Радченко І. В., Шеховцов О. В., Коваленко А. А., Ситник О. В. Формування кластерів на одноплатних комп'ютерах у мережах ІоТ. Системи управління, навігації та зв'язку. Полтава : Національний університет «Полтавська політехніка імені Юрія Кондратюка», 2024. Вип. 2(76). С. 132–136.
2. Резанов, Б., Кучук, Г. Модель розподілу елементарних потоків даних у туманній платформі підтримки інтернету речей. Сучасний стан наукових досліджень та технологій в промисловості. 2023. № 3 (25). С. 88–97. DOI: <https://doi.org/10.30837/ITSSI.2023.25.088>.
3. Petrovska I., Kuchuk H. Static allocation method in a cloud environment with a service model IAAS. Сучасні інформаційні системи. 2022. Том 6, № 3. С. 99-106. DOI: <https://doi.org/10.20998/2522-9052.2022.3.13>
4. Dong Y., Yao Y.-D. IoT Platform for COVID-19 Prevention and Control: A Survey. IEEE Access. 2021. Vol. 9. P. 49929–49941.
5. Javaid M., Khan I.H. Internet of Things (IoT) enabled healthcare helps to take the challenges of COVID-19 Pandemic. J. Oral Biol. Craniofacial Res. 2021. Vol. 11, № 2. P. 209–214.
6. Gupta, N., Juneja, P.K., Sharma, S., Garg, U. An intelligent technique for network resource management and analysis of 5G-IoT smart healthcare application. Journal of Autonomous Intelligence. 2024. 7(1). 694. DOI: [10.32629/jai.v7i1.694](https://doi.org/10.32629/jai.v7i1.694).
7. Eccles, B.J., Rodgers, P., Kilpatrick, P., Spence, I., Varghese, B. DNN Shifter: An efficient DNN pruning system for edge computing. Future Generation Computer Systems. 152. P. 43–54. DOI: [10.1016/j.future.2023.09.025](https://doi.org/10.1016/j.future.2023.09.025)
8. Daru, A.F., Hartomo, K.D., Purnomo, H.D. IPv6 flood attack detection based on epsilon greedy optimized Q learning in single board compute. International Journal of Electrical and Computer Engineering. 13(5). P. 5782–

5791. DOI: 10.1016/j.iot.2023.100764

9. PicoCluster Pico 5 w/Raspberry Pi 4 8GB. Сайт фірми MakerBright. URL: <https://makerbright.com/picocluster-pico-5-w-raspberry-pi-4-8gb.html> PINEPHONE PRO. URL: <https://www.pine64.org/10>. The Beast spotted in the wild. URL: <https://pbs.twimg.com/media/DP9yGqWXUAECqPn.jpg>
10. Oracle представила суперкомп'ютер із тисячі Raspberry Pi 3. URL: <https://www.overclockers.ua/news/hardware/2019-09-18/125252/>
11. ROCKPro64 Cluster Move June 5-10. URL: <https://www.pine64.org/2020/06/05/rockpro64-cluster-move-june-5-10/>
12. Introduction to Azure Services, Stack 24/7, .Net, JavaScript, Azure, Cloud, Coding, Idea and stuff. URL: <https://stack247.wordpress.com/>
13. What is Azure Service Platform? URL: <https://www.netreo.com/cloud-monitoring/what-is-azure-service-platform/>