

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод організації обчислень у
туманному середовищі

(тема)

Виконав:

студент II курсу, групи СПМ-22-5
Колесник З.В.
(прізвище, ініціали)

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

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

Освітня програма Системне програмування
(повна назва освітньої програми)

Керівник: проф. Кучук Г.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Коваленко А.А.

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

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

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

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Колеснику Захару Володимировичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Метод організації обчислень у туманному середовищі

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 _____

Операційна система – Windows, файл формату *.txt, який містить дані,
програмні модулі.

4. Перелік питань, що потрібно опрацювати у роботі _____

- 1) провести аналіз сучасних тенденцій в області організації розподілених обчислень;
- 2) розглянути питання перенесення навантаження до туманного шару в системах IoT;
- 3) розробити метод розподілу навантаження туманного шару IoT та дослідити його ефективність;
- 4) проведення експериментальних досліджень;
- 5) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 14

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)

проф. Кучук Г.А.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 79 с., 27 рис., 1 табл., 3 дод., 18 джерел.

ІНТЕРНЕТ РЕЧЕЙ, ТУМАННИЙ ШАР, ХМАРНЕ СЕРЕДОВИЩЕ,
РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ, ОБЧИСЛЮВАЛЬНИЙ ВУЗОЛ.

Метою кваліфікаційної роботи є підвищення ефективності використання обчислювальних пристроїв туманного середовища Інтернету речей.

У ході виконання кваліфікаційної роботи визначені сучасні тенденції в області організації розподілених обчислень. Розглянуті питання перенесення навантаження до туманного шару в системах Інтернету речей. Удосконалений метод розподілу навантаження туманного шару Інтернету речей за рахунок формування локальних груп пристроїв, що сприяло виграшу за сумарним часом поточного розподілу обчислювальних пристроїв та надало можливість підвищити ефективність виконання завдань.

ABSTRACT

Master's thesis: 79 pages, 27 figures, 1 tables, 3 appendices, 18 sources.

INTERNET OF THINGS, FOG LAYER, CLOUD ENVIRONMENT,
DISTRIBUTED COMPUTING, COMPUTING NODE.

The major goal of this thesis is to improve the efficiency of using computing devices in the foggy environment of the Internet of Things.

In order to modern trends in the field of distributed computing organization were determined. Issues of load transfer to the fog layer in Internet of Things systems are considered. An improved method of distributing the load of the foggy layer of the Internet of Things due to the formation of local groups of devices, which contributed to the gain in terms of the total time of the current distribution of computing devices and provided an opportunity to increase the efficiency of task performance.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 АНАЛІЗ СУЧАСНИХ ТЕНДЕНЦІЙ В ОБЛАСТІ ОРГАНІЗАЦІЇ РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ	12
1.1 Технологія ґрід-обчислень	12
1.2 Технологія хмарних обчислень	15
1.3 Технологія туманних обчислень у розподілених системах	19
2 РОЗГЛЯД ПИТАНЬ ПЕРЕНЕСЕННЯ НАВАНТАЖЕННЯ ДО ТУМАННОГО ШАРУ В СИСТЕМАХ ІНТЕРНЕТУ РЕЧЕЙ	26
2.1 Моделі розпаралелювання завдань Інтернету речей	26
2.2 Варіанти перенесення навантаження до туманного шару	28
2.3 Постановка завдання перенесення навантаження до туманного шару в системах Інтернету речей	31
3 РОЗПОДІЛ НАВАНТАЖЕННЯ ТУМАННОГО ШАРУ ІНТЕРНЕТУ РЕЧЕЙ	38
3.1 Розподіл навантаження у туманному середовищі з використанням локальних груп пристроїв	38
3.2 Методика розміщення навантаження в туманному середовищі	43
3.3 Вибір стратегії опитування вузлів туманного шару	45
3.4 Оцінка ефективності методу розподілу навантаження туманного шару Інтернету речей	52
3.4.1 Залежність часу реалізації розробленого методу від вибору алгоритму обрання лідера	54
3.4.2 Оцінка часу моделювання розподілу навантаження	56
ВИСНОВКИ	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	63
ДОДАТОК А Графічний матеріал кваліфікаційної роботи	66

ДОДАТОК Б Публікація	74
ДОДАТОК В Тексти окремих модулів програмного комплексу.....	76

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

ІР – Інтернет речей
КВ – Координаційний вузол
ЛГП – локальна група пристроїв
Мб – мегабайт
МВ – Майстер-вузол
ОВ – обчислювальний вузол
ОЗП – оперативний запам'ятовуючий пристрій
ОС – операційна система
ПК – персональний комп'ютер
ТД – транзитна ділянка
ХЦОД – хмарний центр обробки даних
ЦОД – центр обробки даних
ЕС – Edge Computing
ІоТ – Internet of Things
VPN – Virtual Private Networks
WCG – World Community Grid

ВСТУП

Значне розширення областей застосування розподілених систем обумовлено безліччю факторів, серед яких є такі:

- необхідність обробляти великі дані в географічно розподіленому середовищі;
- високі вимоги до продуктивності з можливістю масштабованості обчислювальних систем;
- необхідність спільного використання ресурсів;
- виконання вимог до відмовостійкості обчислювальних систем.

Туманні обчислення в даний час є перспективною концепцією організації розподілених обчислень. Вони припускають наближення обробки даних до кінцевих пристроїв мереж. Це дозволяє знизити навантаження на комунікативну мережу і скоротити час відгуку системи. На сьогоднішній день туманна інфраструктура реалізована за допомогою багатоцільових децентралізованих платформ, таких як AWS IoT Greengrass, Azure Functions, SONM тощо.

Розподілені системи, реалізовані на базі технології туманних обчислень, відповідають багатьом вимогам сучасності. Однак в таких системах мають місце часові витрати, пов'язані з процесами реконфігурації системи, які в умовах туманного середовища відбуваються багаторазово, що надає негативного впливу на ефективність функціонування системи.

Ефективність функціонування системи – це властивість, що характеризує ступінь здібності системи виконувати свою функцію за призначенням (досягнення мети), зокрема для туманного середовища одним з параметрів для визначення ефективності є час виконання поточних завдань, який особливо важливий для систем Інтернету речей. Одним із завдань, які вирішуються при організації обчислювального процесу систем Інтернету речей є завдання розподілу обчислювального навантаження на пристроях, наближених до кінцевих датчиків. Вважаючи на відносно велику кількість

запитів на запуск даного завдання скорочення часу його виконання даної дозволяє підвищити ефективність функціонування системи Інтернету речей.

Існує ряд робіт, в яких запропоновані методи розподілу обчислювального навантаження в розподілених системах. Однак туманне середовище підтримки Інтернету речей має особливості, які накладають суттєві обмеження на моделі, методи та алгоритми, які застосовуються для вирішення завдання розподілу обчислювального навантаження. За рахунок цього в певних умовах виконання їх функціональних завдань стає недостатньо ефективним, або навіть неможливим.

Отже, спираючись на вищевикладене, науково-технічне завдання, спрямоване на розробку методу розподілу навантаження туманного шару Інтернету речей є актуальним та доцільним.

Предметом дослідження є моделі та методи розподілу навантаження розподілених систем.

Об'єктом дослідження є процес переносу та розподілу навантаження на туманний шар Інтернету речей.

Метою дослідження є підвищення ефективності використання обчислювальних пристроїв туманного середовища Інтернету речей.

Для досягнення мети повинні бути вирішені такі часткові завдання:

- провести аналіз сучасних тенденцій в області організації розподілених обчислень;
- розглянути питання перенесення навантаження до туманного шару в системах Інтернету речей;
- розробити метод розподілу навантаження туманного шару Інтернету речей;
- дослідити ефективність методу розподілу навантаження туманного шару Інтернету речей.

Наукова новизна дослідження полягає в такому: удосконалений метод розподілу навантаження туманного шару Інтернету речей за рахунок формування локальних груп пристроїв, що сприяло виграшу за сумарним

часом поточного розподілу обчислювальних пристроїв та надало можливість підвищити ефективність виконання завдань.

Практична цінність дослідження полягає в підвищенні ефективності виконання завдань Інтернету речей.

За результатами проведених досліджень була надрукована наукова стаття в фаховому виданні за 123 спеціальністю «Системи управління, навігації та зв'язку» [1].

1 АНАЛІЗ СУЧАСНИХ ТЕНДЕНЦІЙ В ОБЛАСТІ ОРГАНІЗАЦІЇ РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ

1.1 Технологія грід-обчислень

Зародження грід-технологій пов'язано з необхідністю рішення різних прикладних завдань високої обчислювальної складності в різних галузях, таких як наука, медицина, економіка, виробництво, інженерія тощо [2]. Рішення таких завдань на окремих комп'ютерах неможливо було тримати за розумний час. Грід-технології розглядаються як форма розподілених обчислень, в якій «віртуальний суперкомп'ютер» представлений в вигляді кластерів, з'єднаних з допомогою мережі слабпов'язаних різнорідних комп'ютерів, працюючих разом для виконання численних завдань. Грід-технології – це спосіб спільного скоординованого використання розподілених ресурсів. Перевага грід-технологій полягає в можливості проведення паралельних обчислень на неспеціалізованих комп'ютерах, пов'язаних мережею, що значно зменшує їх вартість в порівнянні з суперкомп'ютерами (рисунок 1.1).

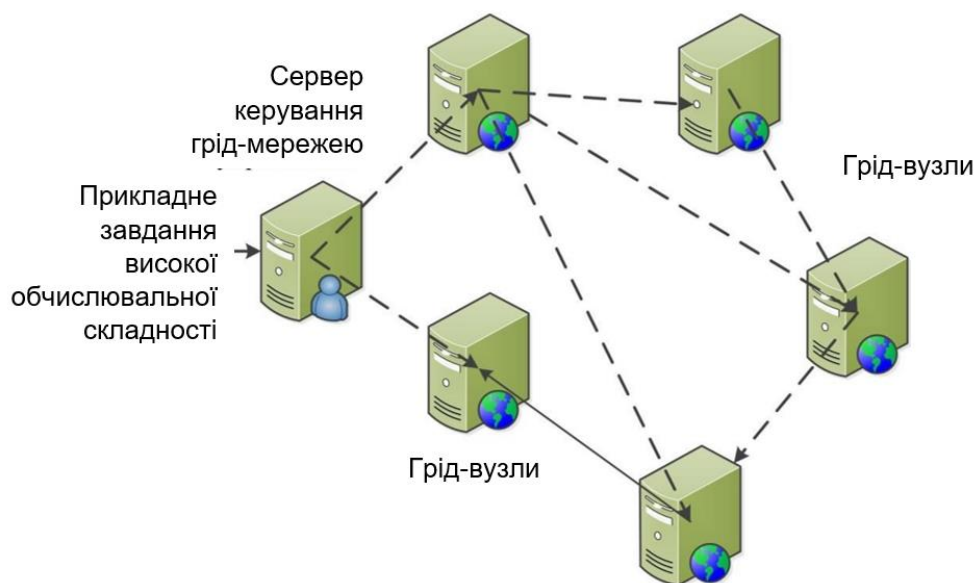


Рисунок 1.1 – Узагальнена архітектура технології грід-обчислень

Розрізняють добровільні гріди, які засновані на добровільному наданні вільних ресурсів ПК, наукові гріди-системи, що реалізують добре розпаралелювані застосунки, та комерційні, функціонуючі на віртуальному комп'ютері, що складається з набору фізичних комп'ютерів, об'єднаних за допомогою технології гріди.

Принципи організації гріди-інфраструктури:

- розподіленість на множині адміністративних доменів;
- сильна гетерогенність;
- розподіленість;
- динамічність, адаптивність;
- віртуальність;
- наявність механізму виявлення вільних ресурсів.

Основні напрямки використання гріди-технологій:

- організація ефективного використання обчислювальних ресурсів тимчасово простоюючих комп'ютерів для рішення відносно невеликих завдань;
- рішення дуже складних обчислювальних завдань, які вимагають величезні процесорні ресурси, ресурси пам'яті тощо;
- обчислення, пов'язані з обробкою великого обсягу географічно розподілених даних;
- колективні обчислення, які мають на увазі участь користувачів з різних організацій або афілій.

Одним з напрямів використання гріди-технологій є їх застосування для проведення глобальних досліджень, які мають на увазі виконання величезних обчислень. World Community Grid (WCG) – глобальна комп'ютерна мережа, що займається гріди-обчисленнями для пошуку рішення глобальних наукових завдань. Колективні обчислення такого роду засновані на ідеї добровільної участі користувачів у процесі проведення глобальних обчислень. Підключення до мережі будь-якого бажаючого можливо за допомогою програми-клієнта, яка буде використовувати обчислювальні ресурси

комп'ютера, котрий не завантажений в даний період часу. Таким чином, World Community Grid можна розглядати як глобальну спільноту користувачів персональних комп'ютерів, які надають невикористовуваний час своїх обчислювальних систем для реалізації глобальних дослідних ініціатив. На даний час у співтоваристві WCG зареєстровано понад 811 тис. користувачів, представлених 8,3 млн. обчислювальними пристроями (рисунок 1.2).

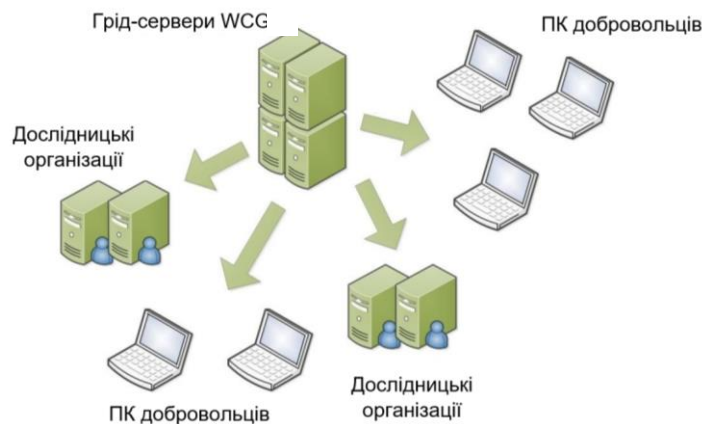


Рисунок 1.2 – Узагальнена схема глобальних досліджень грід-технологій

На сьогодні спільнота WCG проводить низку проектів, серед яких моделювання опадів на африканському континенті, виявлення маркерів, пов'язаних із різними видами раку, пошук методів лікування потенційних пандемій, пошук кращих методів лікування до більшої кількості видів дитячого раку, моделювання поведінки молекул туберкульозу та інші.

Більше того, багато проектів вже є завершеними: проект по вивченню просторової структури білків людського організму і вивченню їх функцій, проект з пошуку нових методів лікування синдрому придбаного імунodefіциту, проект по створенню повної карти білків людського організму, проект по покращення терапії раку на ранньому етапі діагностики, проект по розшифровці інформації геномів різних організмів та інші.

Стоїть відзначити, що в випадках, коли здійснюється географічно розподілений збір даних для подальшої їх обробки у розподіленій системі, то

виникають обмеження, пов'язані з передачею цих даних по каналів зв'язку. Це, у свою чергу, має безпосередній вплив на швидкість відгуку системи, що є однією з найважливіших характеристик розподілених систем.

1.2 Технологія хмарних обчислень

Розвиток грид-технологій призвів до зародження технології «хмарних» обчислень (від англ. cloud computing). Появу хмарної концепції відносять до початку 1950-х років і пов'язують з появою перших мейнфреймів, які надавали одночасний доступ кількох користувачів до центрального комп'ютера. У 1960-х своє розвиток отримала концепція Дж. Ліклайдера «міжгалактичної комп'ютерної мережі», основні ідеї якої подібні з «хмарною» концепцією.

Хмарні обчислення – це модель, що дозволяє легко здійснювати мережний доступ на вимогу до єдиного пулу обчислювальних ресурсів (наприклад, мереж, серверів, систем зберігання даних, застосунків і послуг), які швидко виділяються для використання і потім також швидко вивільняються [3].

Хмарні обчислення почали активно розвиватися і впроваджуватися в різні галузі науки і техніки з 2007 року. Це пов'язано, в основному, з активним розвитком каналів зв'язку і потребою користувачів щодо масштабування автоматизованих систем. Основні елементи хмарної концепції схематично показані на рисунку 1.3.

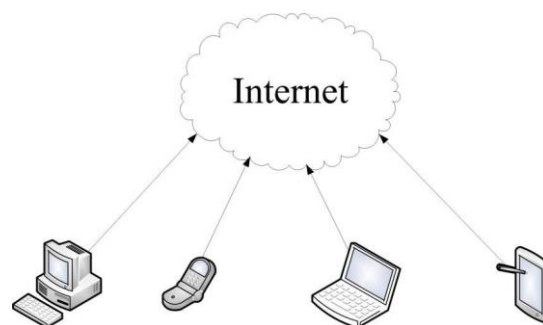


Рисунок 1.3 – Концепція хмарних обчислень

На базі хмарних обчислень функціонує велика кількість різних застосунків. Це, наприклад, поштові послуги, такі як Gmail, Yahoo! Mail, Webmail, Hotmail, онлайнві текстові редактори Zoho Writer, Документи Google, онлайнві сервіси роботи з графікою Lunaric, Google Picasa. На сьогодні хмарна концепція пропонує різноманітні моделі послуг, включаючи такі, як модель «платформа як послуга», модель «програмне забезпечення як послуга», модель «інфраструктура як послуга», модель «дані як послуга» та ін.

Вже більше 10 років існує мультихмарна концепція, яка припускає наявність розподіленої хмарної структури на базі так званих гіперскейлерів – хмарних провайдерів масштабу Google, Amazon або Azure.

Ключові переваги використання хмарних технологій для користувача відображені на рисунку 1.4.

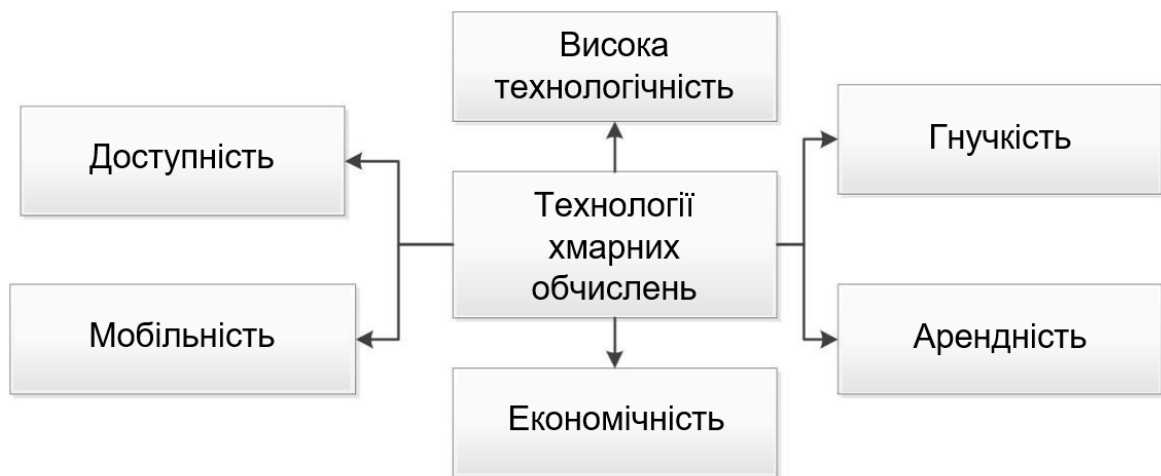


Рисунок 1.4 – Переваги технології хмарних обчислень для користувача

Детально розглянемо переваги хмарних обчислень для користувача, що наведені на рисунку 1.4. Тут, під доступністю розуміється можливість отримання доступу до інформації за наявності будь-якого пристрою, який має підключення до мережі Інтернет. При використанні хмарних технологій зростає мобільність осіб, які беруть участь у розробці будь-якого проекту,

оскільки відпадає необхідність прив'язки працівника до конкретного місця. Економічна ефективність використання даної технології пов'язана з відсутністю необхідності дорогого обслуговування апаратної інфраструктури, а також придбанням ліцензій на використовуване програмне забезпечення. Використання хмарних технологій дозволяє резервувати хмарні центри обробки даних (ЦОД) відповідно до необхідного навантаження, що підвищує ефективність використання обчислювальних ресурсів. При цьому всі необхідні ресурси виділяються провайдером автоматично, що дозволяє говорити о гнучкості використання технології хмарних обчислень. Хмарні ЦОД мають величезну обчислювальну потужність, яку користувач може використовувати для рішення завдань, пов'язаних з зберіганням, аналізом та обробкою даних.

Перераховані вище особливості дозволяють говорити про широку область застосування технології хмарних обчислень.

Однак поряд з очевидними перевагами хмарної та мультихмарної концепцій, як відзначають спеціалісти, існує ряд обмежень їх застосування в області розподілених систем, до яких відносяться такі [4]:

- затримки в мережі при передачі даних, пов'язані з територіальним розташуванням хмари і граничних пристроїв (наприклад, при 3D-моделюванні);

- затримки, пов'язані з обробкою даних в зв'язку з великим обчислювальним навантаженням на хмарний сервер (наприклад, обробка графіки, розрахунок схем високого ступеня інтеграції тощо).

Більше того, специфіка функціонування розподілених систем полягає у нерівномірності споживання обчислювальних ресурсів. Це пов'язано з тим, що комплексні функціональні завдання розподілених систем мають різну обчислювальну складність на різних етапах проектування та виконуються колективом розробників паралельно, що призводить до можливості виникнення пікового навантаження у мережі, що, своєю чергою, може призвести до такого:

- неможливість отримання прийняттого за якістю проектного рішення на наявних потужностях за відведений час;
- отримання прийняттого за якістю проектного рішення на наявних потужностях за відведений час, але неможливість його покращення, що суперечить головному принципу ефективного функціонування більшості розподілених систем автоматизованого проектування.

Дані досліджень показали, що при традиційному підході до створення IT-інфраструктури для уникнення вище описаних ситуацій, замовник зазвичай замовляє під майбутні навантаження додатково в середньому 59% обчислювальних потужностей і 48% ємностей систем зберігання даних.

З іншого боку, через раптові та непередбачені пікові навантаження 57% компаній періодично відчують труднощі через нестачу обчислювальних потужностей, а великий відсоток опитуваних мали досвід зупинки IT-систем, що не справляються з поточними навантаженнями [5].

Перелічені проблеми можуть бути вирішені за рахунок залучення додаткових обчислювальних потужностей, у тому числі за рахунок резервування хмарних дата центрів. Прикладом такого рішення є модель HPE GreenLake, запропонована американською компанією Hewlett Packard Enterprise . Головною особливістю даної моделі є можливість при необхідності розмістити частину сервісів HPE GreenLake і відповідних апаратних рішень на зовнішній локації: у публічній або приватній хмарі, зовнішньому ЦОД або на обладнанні, фізично розташованому на території замовника. Дана модель, з однієї сторони, дозволяє компаніям не переплачувати за обладнання, що часто купується «із запасом». І, з іншого боку, такий підхід дає можливість не турбуватися про нестачу обчислювальних ресурсів для повсякденної діяльності [6].

Слід також відзначити, що на сьогоднішній день існує ряд робіт, що демонструють методи рішення завдань розподілу обчислювального навантаження та отримання додаткових потужностей у розподілених системах за рахунок використання пристроїв туманного шару [7].

1.3 Технологія туманних обчислень у розподілених системах

В даний час нова парадигма туманних обчислень, яку можна розглядати як розширення хмарної концепції, знайшла широке застосування у багатьох областях. Це пов'язано з тим, що на сьогоднішній день хмари не відповідають високим вимогам мобільності, низької латентності та локальної поінформованості [8]. Перспективним рішенням в даному випадку бачиться вибіркове зміщення обчислень, зв'язку, управління і прийняття рішень до місць, де генеруються дані, що відповідає основним принципам концепції туманних обчислень.

Концепція туманних обчислень є еволюційним етапом розвитку хмарної концепції і в даний час займає лідируючу позицію серед загальних тенденцій розвитку інформаційних технологій. Поява даної концепції тісно пов'язано з зародженням і розвитком концепції «Інтернету речей» [9]. Сценарії використання концепції туманних обчислень різноманітні і обумовлюються розвитком суміжних технологій. Дана концепція успішно використовується у створенні систем типу «розумний» будинок, «розумний» транспорт», електронне охорона здоров'я, електронний уряд, торгівля і фінансові послуги, промислове виробництво, управління технологічними і бізнес процесами і багато інших.

Термін «туманні обчислення» вперше введений компанією Cisco в 2011 році і на поточний момент активно розвивається завдяки наступним факторам:

- обчислювальні потужності комунікаційного обладнання і пристроїв, розміщених у туманному шарі, дозволяють виконувати додаткові обчислення;
- кількість кінцевих пристроїв зростає дуже швидко, і ця тенденція збережеться в доступній для огляду перспективі;
- доцільно розвантажити ЦОД (зокрема хмарні) від виконання якоїсь частини складних обчислювальних процесів [10].

Туманні обчислення – це багаторівнева модель, що забезпечує повсюдний доступ до загальної сукупності обчислювальних ресурсів, що масштабуються. Туманні обчислення мінімізують час мережного відгуку підтримуваних застосунків, а також забезпечують кінцеві пристрої локальними обчислювальними ресурсами і, при необхідності, мережним підключенням до централізованих сервісів. Архітектуру туманних обчислень можна розглядати як «прошарку» між хмарою і кінцевими пристроями користувача (рисунок 1.5).

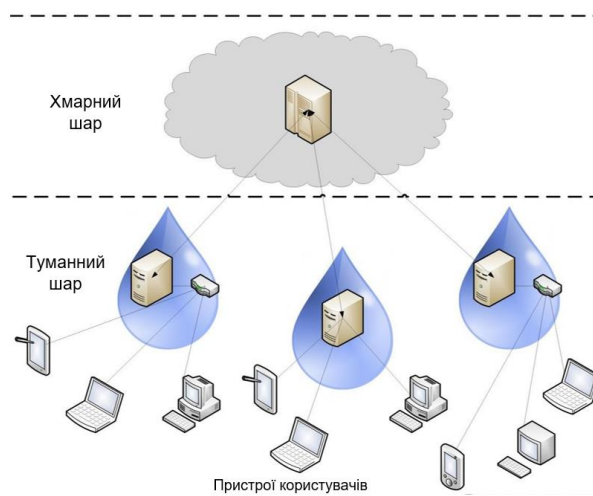


Рисунок 1.5 – Концепція туманних обчислень

Середовище туманних обчислень складається з таких компонентів мережі, як роутери, комутаційні пристрої, телевізійні перетворювачі, проксі-сервери, базові станції, датчики тощо.

Ключова відмінність туманної концепції від хмарної полягає в можливості динамічного перенесення обчислювального навантаження з хмари на периферію мережної інфраструктури із задіянням пристроїв туманного шару, а також часткове розміщення навантаження в туманному шарі. Це дозволяє суттєво знизити навантаження на комунікаційну інфраструктуру мережі [8].

Основні проблеми, котрі пов'язані з організацією концепції Інтернету речей, і застосуванням технології туманних обчислень для їхнього вирішення, відображені на рисунку 1.6.



Рисунок 1.6 – Перспективи використання технології туманних обчислень для вирішення проблем, пов'язаних з організацією Інтернету речей

Великі хмарні провайдери Amazon, Google і Microsoft ведуть розробки по впровадженню концепції туманних обчислень на базі безсерверної архітектури (serverless architecture). Компанія Amazon пропонує платформу туманних обчислень AWS IoT Greengrass, яка ефективно розширює можливості хмарної інфраструктури на пристрої туманного шару, це дозволяє обробляти дані локально, використовуючи при цьому хмару для управління, аналізу і зберігання даних. Така платформа дозволяє програмними засобами фільтрувати дані пристроїв та передавати назад до хмари тільки необхідну інформацію.

Компанія Microsoft запропонувала рішення для безсерверних обчислень Azure Functions, яке дозволяє запускати невеликі фрагменти програмного коду або функцій в хмарі. Функції Azure забезпечують обробку даних, інтеграцію систем, роботу з Інтернетом речей (IoT) і побудову простих API-інтерфейсів.

Google представила платформу для Інтернету речей Android Things з підтримкою мікрокомп'ютерів Intel Edison та Joule [™] 570x, NXP Pico і.MX6UL та Argon і.MX6UL, а також Raspberry Pi 3.

На сьогоднішній день представлена велика різноманітність туманних платформ, серед них приватні (Cisco IO, Nebbiolo Technologies, ClearBlade, Smartiply Fog, LoopEdge), публічні (Azure IoT, Amazon AWS IoT Greengrass, Google, Yandex) і платформи з відкритим виконавчим кодом (FogFrame 2.0, FogFlow, FogBus).

Ключова особливість технологій туманних обчислень, яка полягає в виконанні більшої частини обробки даних на «границі» мережі, дозволяє їх використовувати в широкій номенклатурі розподілених систем, в тому числі в системах моніторингу і управління, де час відгуку системи є одним з основоположних показників.

Окремо розглянемо системи моніторингу і прогнозування. У останні десятиліття необхідність врахування відомостей про стан навколишнього середовища значно зростає. Насамперед це пов'язано з підвищенням вимог до забезпечення екологічної безпеки, що, у свою чергу, є запорукою стійкого розвитку суспільства. Інтенсивний вплив антропогенних факторів на довкілля може викликати різні наслідки, що виражаються, в тому числі, у негативних природних явищах, які становлять небезпеку для населення і різних об'єктів інфраструктури. У зв'язку з цим, актуальним є використання систем моніторингу, які забезпечують своєчасний збір даних з метою оповіщення населення про передбачуваний наступ небезпечних подій. Усе різноманіття існуючих в даний час систем моніторингу може бути розділеним на системи, які не здійснюють обробку даних, і на системи, які здатні виробляти попередню обробку даних, тим самим підвищуючи функціональність системи. Перспективним напрямком створення систем моніторингу є реалізація такого класу систем на базі технологій "цифрової економіки".

Наприклад, в роботі [11] розглянута проблема розробки системи моніторингу і прогнозування небезпечних природних явищ, яка забезпечує обробку великих обсягів неструктурованих даних, згенерованих гетерогенними пристроями. Запропоновано використовувати технологію

туманних обчислень для розміщення деяких обчислювальних завдань на вузлах мережного обладнання та мобільних пристроїв організацій і приватних осіб, які задіяні в проведенні моніторингу. Дана технологія дозволяє знизити навантаження на комунікаційну мережу і частково розвантажити центри обробки даних, що, в свою чергу, припускає підвищення надійності системи та знижує її латентність.

Поява технологій Інтернету речей та розвиток на їх основі «розумних» міст, територій, підприємств і будинків також сприяє активному впровадженню систем моніторингу в їхню інфраструктуру. У роботах [12, 13] авторами запропонована ієрархічна розподілена архітектура туманних обчислень для підтримки інтеграції великого числа компонентів інфраструктури і сервісів в «розумних» містах. У якості прикладу, демонструючого ефективність пропонованої архітектури, був реалізований прототип системи інтелектуального моніторингу трубопроводів в «розумних» містах. На підставі отриманих результатів можна зробити висновок щодо перспективності використання туманних технологій в інфраструктурі «розумних» міст, що дозволяють значно підвищити їхню "інтелектуальність".

У роботі [14] запропоновано архітектуру інтелектуального розподілу води і моніторингу стану підземних трубопроводів з підтримкою Інтернету речей для «розумних» міст. Ключовим компонентом розробленої архітектури є інтеграція технологій Інтернету речей з туманною і хмарною концепцією, що дозволяє ефективно вирішувати проблеми, пов'язані з автоматизованим розподілом води, а також здійснювати моніторинг витоків трубопроводів. Системи моніторингу активно використовують у сфері сільського господарства. Наприклад, у роботі [15] запропоновано підхід до аналізу та обробки даних для розподіленого моніторингу сільськогосподарських культур та ґрунту, в якому компоненти, що реалізують ієрархічний збір і моделювання даних, дозволяють уникнути проблем, пов'язаних з обмеженням пропускної спроможності мережі і скоротити енергію,

необхідну для передачі даних. Тут головна увага приділяється застосуванню технології туманних обчислень, що дозволяє використовувати обчислювальні ресурси локальних вузлів для подальшої передачі частково оброблених даних вищим рівням системи з метою ухвалення рішень.

Таким чином, можна, можливо зробити висновок про те, що побудова систем моніторингу на базі технологій туманних обчислень перспективна і ефективна з точки зору скорочення навантаження на комутаційну мережу, зниження латентності системи та часткового розвантаження центрів обробки даних для різних сфер діяльності людини, про що свідчить проведений вище аналіз.

Ще проведемо більш детальний огляд інформаційно-керуючих систем. В даний час управління складними технічними об'єктами та системами неможливо без використання інформаційно-керуючих систем. Розвиток інформаційних і мережних технологій дозволяє реалізовувати інформаційно-керуючі системи на базі використання хмарної концепції. Однак у системах з високими вимогами до рівня безпеки, високою критичності таких систем до відмов, передача великих обсягів даних з метою їх подальшої обробки або зберігання в хмарні сервери не завжди є доцільним. Вирішенням цієї проблеми є реалізація інформаційно-керуючих систем на базі концепції туманних обчислень, які дозволяють виробляти значну частина обробки даних в мережі між граничним пристроєм і хмарним сервером. У рамках туманної концепції всі пристрої мережі розташовуються в трьох шарах: граничному (користувацькі), туманному (різні комунікаційні пристрої), хмарному (ЦОД). Реалізація інформаційно-керуючих систем на базі концепції туманних обчислень дозволяє вирішити наступні завдання:

- суттєво знизити навантаження на мережу;
- розвантажити хмарні сервера;
- низити латентність системи.

У роботі [16] запропоновано модель розподілу завдань за обчислювальними вузлами з урахуванням географічної розподіленості

системи та накладних витрат на передачу даних через мережу. Авторами були досліджені області застосування концепції туманних обчислень для ефективного розподілу обчислювального навантаження в розподілених системах з точки зору підвищення їх надійності. Результати експерименту показали, що підвищення надійності можливо, але в певних умовах, тому для уточнення області застосування даної концепції необхідно проводити моделювання розміщення обчислювального навантаження.

У роботі [17] розглянута доцільність використання технології туманних обчислень при розробці інформаційно-керуючих систем з точки зору надійності. Представлено модель інформаційно-керуючої системи, реалізованої з використанням технологій розподіленого реєстру та концепції туманних обчислень. Основний результат дослідження складається в отриманому комплексі аналітичних оцінок, на підставі якого розміщення копій реєстру в туманному шарі є доцільним з точки зору надійності системи. Однак із зростанням обчислювального навантаження можливість доцільного розміщення реєстру в туманному шарі обмежується малою продуктивністю туманних вузлів.

Однак, незважаючи на явні переваги концепції туманних обчислень, які активно використовуються при створенні широкої номенклатури розподілених систем, дана концепція не позбавлена проблем, що вимагають окремих рішень. Одна з таких проблем пов'язана з необхідністю багаторазового виконання завдання перенесення обчислювального навантаження в процесі функціонування системи, що є наслідком динамічності і нестабільності туманного середовища. Рішення завдання реконфігурації безпосередньо пов'язано з часовими витратами, які негативно впливають на ефективність функціонування системи. Таким чином, для підвищення ефективності функціонування системи, необхідно скоротити дані витрати, тобто вирішити завдання перенесення обчислювального навантаження за мінімально можливий час, що особливо суттєво при реалізації географічно розподілених систем.

2 РОЗГЛЯД ПИТАНЬ ПЕРЕНЕСЕННЯ НАВАНТАЖЕННЯ ДО ТУМАННОГО ШАРУ В СИСТЕМАХ ІНТЕРНЕТУ РЕЧЕЙ

2.1 Моделі розпаралелювання завдань Інтернету речей

Існують різні моделі розпаралелювання завдань Інтернету речей. Розглянемо кожну з них докладніше.

Глобальна модель паралелізму є паралельним аналогом своїх послідовних прототипів. У такій моделі організовані паралельні обчислення на кшталт «майстер-підлеглий» (master – slave) (рисунок 2.1).

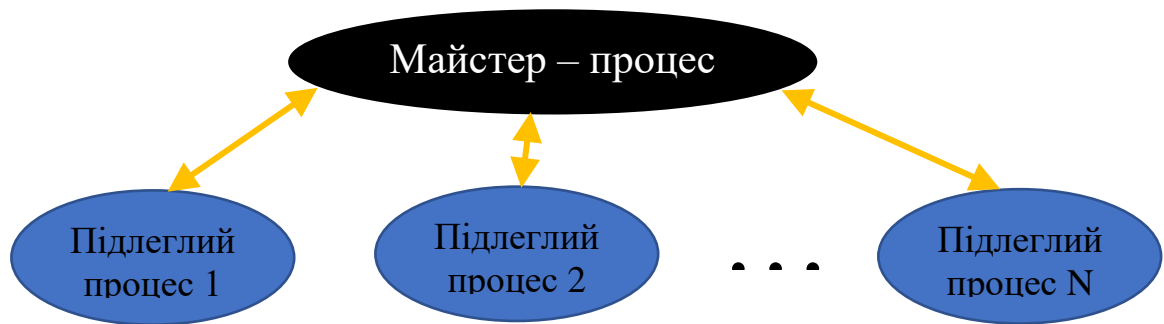


Рисунок 2.1 – Модель «Майстер-підлеглий»

Майстер-процес реалізує завдання Інтернету речей на host-процесорі, робочі процеси виконуються на якихось процесорах паралельних обчислювальних вузлів. Їх завданням є обчислення значень цільових функцій та фазових координат відповідного агента, а також відправка значень цільової функції після кожної ітерації майстер-процесу.

В острівній моделі паралелізму мультизавдання складається з підзавдань або островів, кожен з яких виконується на одному з процесорів паралельних обчислювальних вузлів. Острови обмінюються даними після кожної ітерації відповідно з топологією сусідства островів (рисунок 2.2). сезоном називається період часу, коли кожен острів працює незалежно від іншого і обміну даними між ними не існує.

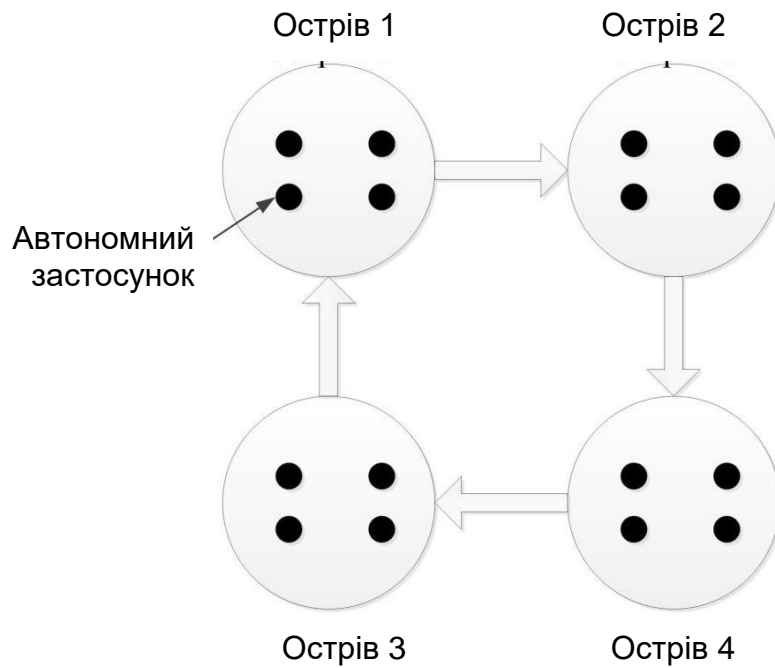


Рисунок 2.2 – Концепція острівної моделі

Існують різні види острівних моделей в залежності від використовуваної топології сусідства, тривалості сезону, стратегії та режимів обміну даними між островами. Наприклад, острівна модель може бути використана для запуску різних варіантів рішення оптимізаційних завдань паралельно на кожному з островів при необхідності вибору з отриманих рішень кращого.

Методи, що використовуються в дифузійній моделі паралелізму (рисунок 2.3), можна віднести до окремого випадку методу, заснованого на острівній моделі.

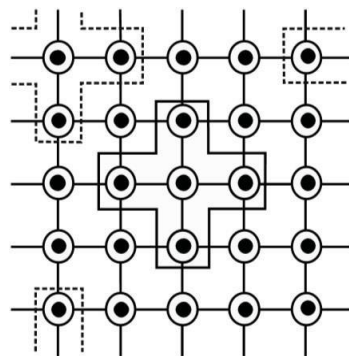


Рисунок 2.3 – Дифузійна модель паралелізму

Кожне підзавдання включає тільки одного агента, так, що число островів дорівнює числу агентів і одночасно числу використовуваних процесорів обчислювальної системи.

Ідея дифузійної моделі розпаралелювання полягає в тому, що обробка окремих кластерів Інтернету речей відбувається паралельно, причому серед сусідніх агентів. Комунікаційні витрати в дифузійній моделі залежать від топології сусідства агентів і є високими в випадку високої зв'язності графа.

Відмітимо, що у разі невисокої пропускної спроможності мережі паралельних обчислювальних вузлів та невисокій обчислювальній складності компонент острівна модель має переваги над глобальною моделлю. У випадку невисокої зв'язності графа та/або високої обчислювальної складності раціональніше використовувати дифузійну модель.

2.2 Варіанти перенесення навантаження до туманного шару

Розглянемо обчислювальне завдання Інтернету речей, що виконуються у хмарному середовищі та можуть бути розпаралелені з метою виконання окремих підзавдань у туманному середовищі. Надалі будемо розглядати граф такого обчислювального завдання. Під способом розбиття графа обчислювального завдання будемо розуміти перелік деяких правил, згідно яких відбувається «відрив» фрагмента графа від заданого. Такі обмеження дозволяють виключити з розгляду безліч варіантів розбиття, отриманих в результаті довільного розбиття графа. На основі моделей розпаралелювання, розглянутих у попередньому пункті, сформулюємо такі правила, згідно з якими граф обчислювального завдання підлягає розбиттю з метою перенесення частини обчислювального навантаження на туманні вузли.

Умовимося, що розбиття вхідного графа обчислювального завдання можна проводити тільки на рівні процесів, відбивають інформаційні обміни в кожній з розглянутих паралельних моделей. У випадку моделі «Майстер-підлеглий» можливі такі варіанти перенесення підграфа завдань:

– перенесення майстер-процесу (рисунок 2.4);

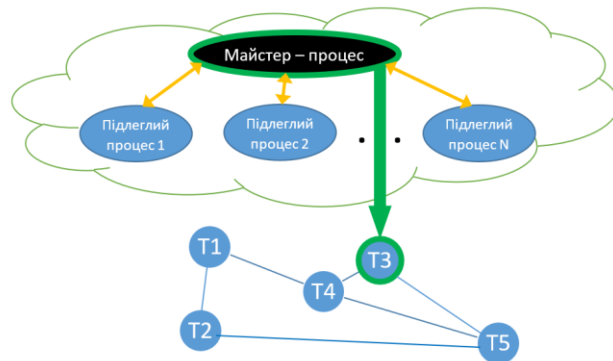


Рисунок 2.4 – Перенесення майстер-процесу на вузол в туманному шарі

– перенесення підлеглого процесу (рисунок 2.5);

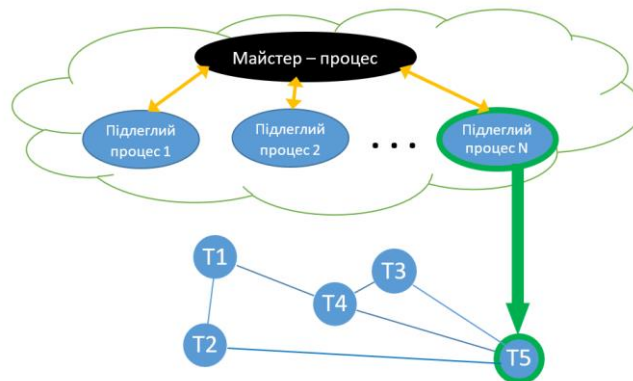


Рисунок 2.5 – Перенесення підлеглого процесу на вузол в туманному шарі

– перенесення майстер-процесу та підлеглого процесу (рисунок 2.6);

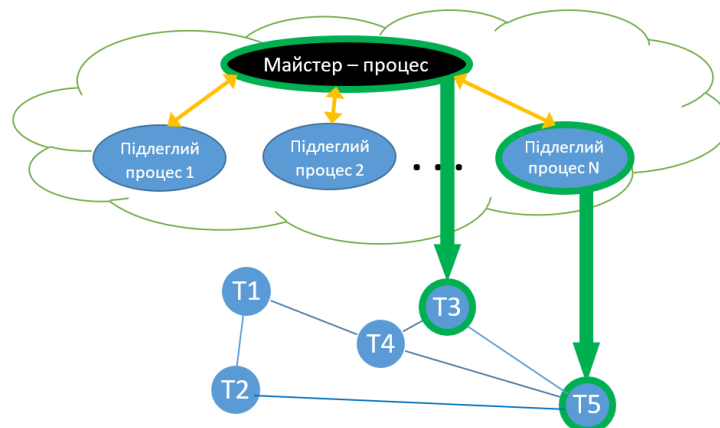


Рисунок 2.6 – Перенесення майстер-процесу та підлеглого процесу до туману

– перенесення декількох підлеглих процесів (рисунок 2.7);

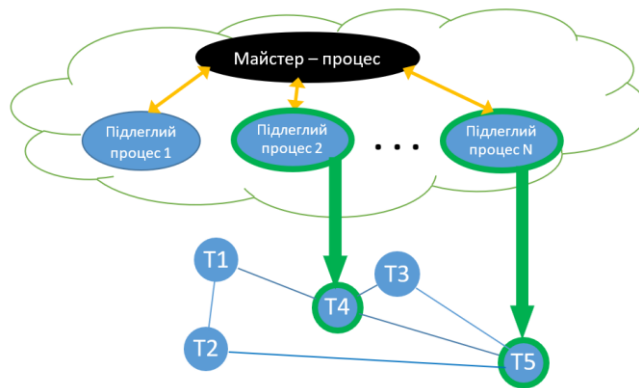


Рисунок 2.7 – Перенесення декількох підлеглих процесів до туману

– перенесення майстер-процесу та декількох підлеглих процесів (рисунок 2.8).

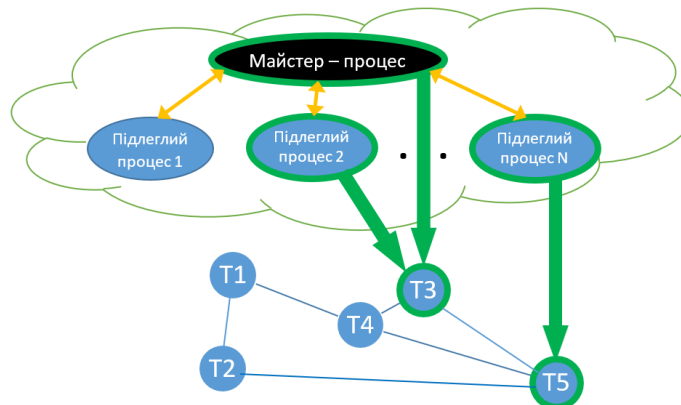


Рисунок 2.8 – Перенесення майстер-процесу та декількох підлеглих процесів

Після розгляду п'яти основних випадків застосування моделі «Майстер-підлеглий» для розпаралелювання завдань Інтернету речей між хмарним та туманним шаром розглянемо варіанти застосування для розпаралелювання завдань острівну модель паралелізму.

В даній ситуації достатньо буде розглянути всього два можливих випадки розбиття графа обчислювального завдання на рівні процесів:

По-перше розглянемо випадок перенесення одного процесу типу «острів» на туманний вузол (рисунок 2.9).

Але у більшості випадків перенесенню підлягає декілька процесів згідно прикладу, наведеному на рисунок 2.10.

Алгоритми, засновані на дифузійній моделі паралелізму, є окремим випадком острівної моделі, тому варіанти їх розбиття можна виключити з подальшого розгляду.

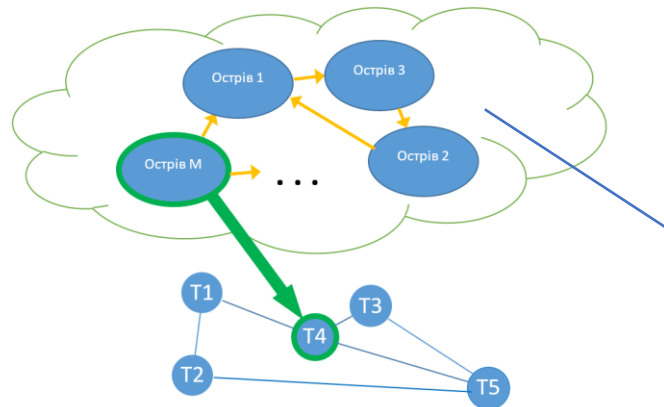


Рисунок 2.9 – Перенесення одного острова до туману

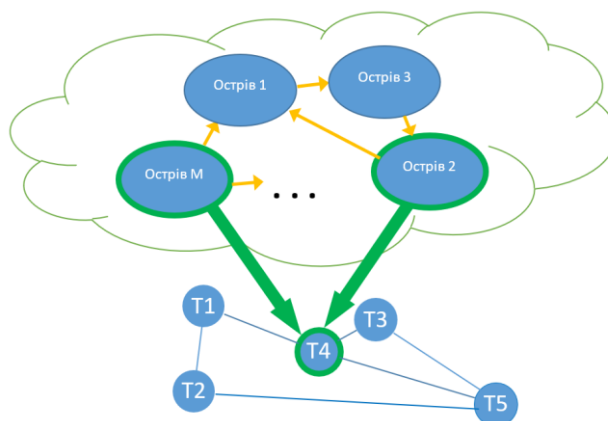


Рисунок 2.10 – Перенесення декількох островів до туману

2.3 Постановка завдання перенесення навантаження до туманного шару в системах Інтернету речей

На сьогодні відома велика кількість методів і алгоритмів розв'язання задачі перенесення та розподілу обчислювального навантаження. Однак більшість із них має суттєві обмеження у своєму використанні.

По перше, не усі методи враховують наявність транзиторних (тимчасових транзитних) ділянок, що є однією з ключових характеристик туманного шару.

По-друге, багато методів орієнтовані на рішення завдань відносно невеликої розмірності і не підходять для складних топологій та гетерогенних середовищ.

По-третє, використовувані для рішення такого роду завдань евристичні алгоритми, їх модифікації і різні комбінації мають високу обчислювальну складність, яка потребує значних часових витрат, що неприпустимо рівною мірою як для умов туманного середовища через його нестабільність і динамічність.

Отже, необхідно розроблення методу, що дозволяє розподілити навантаження завданнями Інтернету речей туманного шару, враховуючи специфіку як туманного шару так і завдань, що до нього надходять.

Розглянемо формальну постановку завдання перенесення обчислювального навантаження з урахуванням наявності тимчасових транзитних ділянок мережі, що властиво туманним середовищам, розробляючи при цьому математичну модель завдання перенесення навантаження з хмари на вузли туманного шару.

Нехай заданий граф обчислювального завдання G , який має для i -го підзавдання деяку обчислювальну складність x_i а обсяг інформації, що передається між підзавданнями дорівнює w_i .

Граф G поділено на два підграфи G' і G'' .

Необхідно розмістити підграф G' на обчислювальні пристрої сегмента мережі P' туманного шару, у той час, якщо обчислювальні завдання підграфа G'' продовжують виконуватись на сегменті мережі P'' (рисунок 2.11).

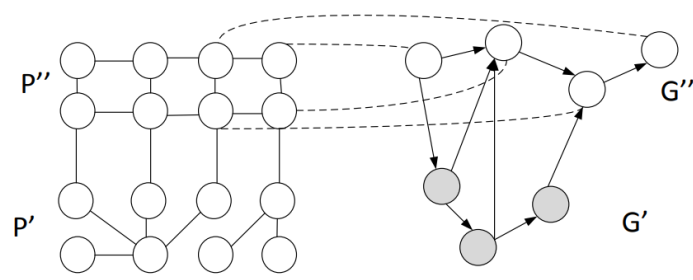


Рисунок 2.11 – Розміщення обчислювальних підзадач при розвантаженні хмарного шару

Відповідно, таке перенесення обчислювальних підзадач припускає рішення оптимізаційної завдання з певними критеріями і обмеженнями, які частково формуються вимогами самої обчислювальної системи. Основним критерієм при рішенні завдання перенесення обчислювального навантаження зазвичай є надійність системи.

Розглянемо графовий опис набору завдань

$$G = \{ \langle i, x_i, w_i \rangle \}, \quad (2.1)$$

де i – унікальний ідентифікатор обчислювального підзавдання, x_i – обчислювальна складність i -го підзавдання, w_i – об'єм інформації, переданий i -м -ой підзавданням до комунікаційного середовища.

Обчислювальні підзавдання графа G пов'язані з вузлами обчислювальних пристроїв, які складають множину P . При цьому множину P можна описати такою графовою структурою

$$P = \{ \langle j, p_j \rangle, list \}, \quad (2.2)$$

де j – унікальний ідентифікатор вузла, p_j – продуктивність j -го вузла, $list$ – матриця поточної пропускнуої здатності каналів зв'язку між інцидентними вузлами мережі.

Тепер розглянемо підграф обчислювальних підзадач G' , які потрібно перенести в той час, коли обчислювальні підзавдання підграфа G'' повинні продовжувати своє виконання. Відмітимо, що поміж даними підграфами існує декілька інформаційних потоків. Потоки, що пов'язані з підграфом G' , можуть бути описані набором таких кортежів:

$$Flow_in = \left\{ \left\langle id_{out}, id_{in}, w_{out_in} \right\rangle \right\}, \quad (2.3)$$

де $\left\langle id_{out}, id_{in}, w_{out_in} \right\rangle$ – кортеж, що описує кількість інформації w_{out_in} , що переданеться між вузлом підмережі G' id_{out} і вузлом підмережі G'' id_{in} в напрямку від G' до G'' :

$$Flow_out = \left\{ \left\langle id_{in}, id_{out}, w_{in_out} \right\rangle \right\}, \quad (2.4)$$

де $\left\langle id_{in}, id_{out}, w_{in_out} \right\rangle$ – кортеж, що описує кількість інформації w_{in_out} , що переданеться між вузлом підмережі G'' id_{in} і вузлом підмережі G' id_{out} в напрямку від G'' до G' .

Тепер, використавши введені позначення, розглянемо формалізацію задачі перенесення обчислювального навантаження з урахуванням наявності тимчасових транзитних ділянок мережі, що властиво туманним середовищам.

Нехай є підграф обчислювальних завдань G'' , пов'язаний з підграфом P'' , а також визначені інформаційні потоки $Flow_in$ та $Flow_out$.

Необхідно розмістити обчислювальні підзавдання підграфа G' на підмножині пристроїв мережі P' таким чином, щоб загальний час виконання обчислювальних підзадач G був менше заданого часу T з урахуванням виконання критерію надійності системи. Рішенням поставленого завдання є встановлення зв'язку між обчислювальними підзавданнями G' і

обчислювальними вузлами підграфа P' , який може бути описаний матрицею A :

$$A = \begin{pmatrix} \langle t_0^{(1,1)}, u_{1,1} \rangle & \dots & \langle t_0^{(1,M)}, u_{1,M} \rangle \\ \dots & \langle t_0^{(i,j)}, u_{i,j} \rangle & \dots \\ \langle t_0^{(N,1)}, u_{N,1} \rangle & \dots & \langle t_0^{(N,M)}, u_{N,M} \rangle \end{pmatrix}, \quad (2.5)$$

де $t_0^{(i,j)}$ – момент часу, коли починається обчислення i -го підзавдання j -м вузлом, $u_{i,j}$ – частка загальної продуктивності p_j заданого j -го вузла для виконання i -го підзавдання.

Дана модель завдання дозволяє пов'язати більше однієї задачі, що приходиться на один вузол одночасно, таким чином, покращена класичну модель формування виконання завдань Інтернету речей.

Для подальшої розробки моделі необхідно розглянути такі параметри:

$L_p(A)$ – завантаження вузла, що породжене перенесенням обчислювального підзавдання на вузол;

$L_{dist}(A, Flow_in, Flow_out)$ – завантаження вузла, що породжене передачею інформації між підграфами G' і G'' ;

$L_{tr}(A, Flow_in, Flow_out)$ – завантаження вузла, що породжене передачею інформації через вузол;

$D_{lk}(A)$ – перелік ребер графа P , готрий визначає маршрут між вузлами l та k ;

$List D_{lk}$ – матриця, що описує пропускну здатність каналів зв'язку між вузлами l та k .

Визначимо цільову функцію як набір значень функцій надійності вузлів:

$$F_j = e^{-\lambda_j t}, \quad (2.6)$$

де λ_j – інтенсивність відмов j -го вузла; t – час функціонування пристрою.

Оскільки

$$\lambda = \lambda_0 \cdot 2^{0,1 \cdot \Delta T}, \quad (2.7)$$

$$\Delta T = k \cdot L, \quad (2.8)$$

де L – завантаження пристрою, k – коефіцієнт, котрий залежить від типу пристрою.

Тоді залежність між функцією надійності і завантаженням описується такою формулою:

$$F_j = e^{-\lambda_0 \cdot 2^{0,1 \cdot k \cdot l}}, \quad (2.9)$$

Завантаження пристрою залежить від розподілу обчислювальних завдань по обчислювальним вузлам, які описуються матрицею A . Розглянуті вище параметри повинні бути включені до моделі задачі:

$$M_G = \langle L_p(A), L_{dist}(A, Flow_in, Flow_out), L_{tr}(A, Flow_in, Flow_out) \rangle, \quad (2.10)$$

Таким чином, повна завантаження j -го обчислювальній вузла описується наступною формулою:

$$L_j = L_{p,j}(A) + L_{dist,j}(A, Flow_in, Flow_out) + L_{tr,j}(A). \quad (2.11)$$

У випадку набору пристроїв, скалярна цільова функція переходить у вектор.

Головним обмеженням для цього завдання є час виконання T графа обчислювального завдання G , тобто:

$$G = G' \cup G'';$$

$$\forall i \in G: \frac{x_i}{p_j \cdot u_{ij}} + t_{dist}(i) < T, \quad (2.12)$$

де $t_{dist}(i)$ – максимальний час надходження інформації від i -го підзавдання до підзавдання-одержувача даних.

Оскільки модель розглядає маршрути потоків інформації, то $t_{dist}(i)$ можна описати функцією:

$$t_{dist}(i) = \xi(A, G, P). \quad (2.13)$$

Більш точно час доставки інформації може бути обчислений на основі повної інформації про підзавдання, призначені на обчислювальні вузли із врахуванням параметрів $D_{lk}(A)$ та $List D_{lk}$.

3 РОЗПОДІЛ НАВАНТАЖЕННЯ ТУМАННОГО ШАРУ ІНТЕРНЕТУ РЕЧЕЙ

3.1 Розподіл навантаження у туманному середовищі з використанням локальних груп пристроїв

У даному підрозділі буде розглянуто метод на основі локальних груп пристроїв (ЛГП), оскільки він враховує більшу частину характеристик, властивих туманному середовищу.

Даний метод полягає в формуванні обмежень на множину обчислювальних пристроїв в режимі реального часу з використанням поняття «локальна група пристроїв».

Під терміном «локальна група пристроїв» розуміється множина пристроїв, які мають такі властивості:

- пристрої взаємопов'язані між собою високошвидкісними каналами зв'язку без транзиторних вузлів;
- пристрої вирішують підзавдання загального обчислювального завдання G ;
- граничні пристрої, що входять до складу однієї ЛГП, можуть одночасно належати сусіднім ЛГП.

Розглянемо постановку завдання.

Нехай заданий граф обчислювального завдання G , котрий поділений на два підграфи G' та G'' . Відповідно до сформульованого у попередньому розділі завдання, необхідно розмістити підграф G' на обчислювальних пристроях туманного шару P' , й одночасно обчислювальні завдання підграфа G'' повинні продовжувати виконуватися на обчислювальних вузлах хмарного шару P'' (рисунок 3.1).

Метод, заснований на формуванні локальних груп пристроїв, складається з виконання кроків розглянутих нижче.

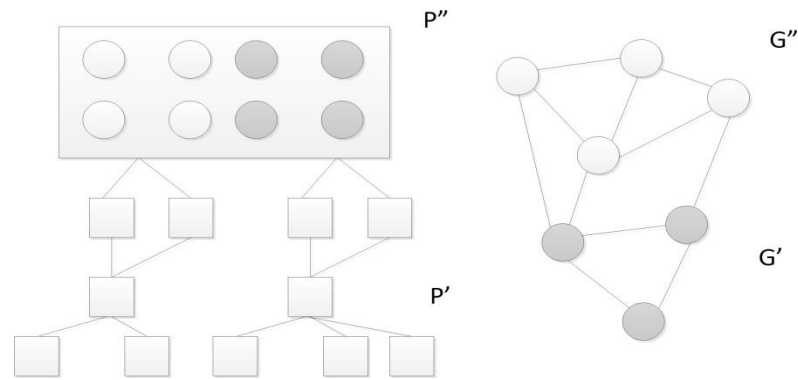


Рисунок 3.1 – Схема перенесення частини обчислювальної навантаження

Крок 1. Оскільки підграф обчислювальних завдань G' необхідно перенести в «туманний» шар, в хмарному шарі повинен бути вузол, відповідальний за перенесення навантаження. Вибір такого вузла-лідера проводиться серед вузлів, виконуючих загальне обчислювальне завдання, за допомогою якогось алгоритму вибору лідера, наприклад, алгоритм Фінна.

Крок 2. Вибраний вузол-лідер опитує свою ЛГП про наявні ресурси для розміщення додаткового навантаження (рисунок 3.2).

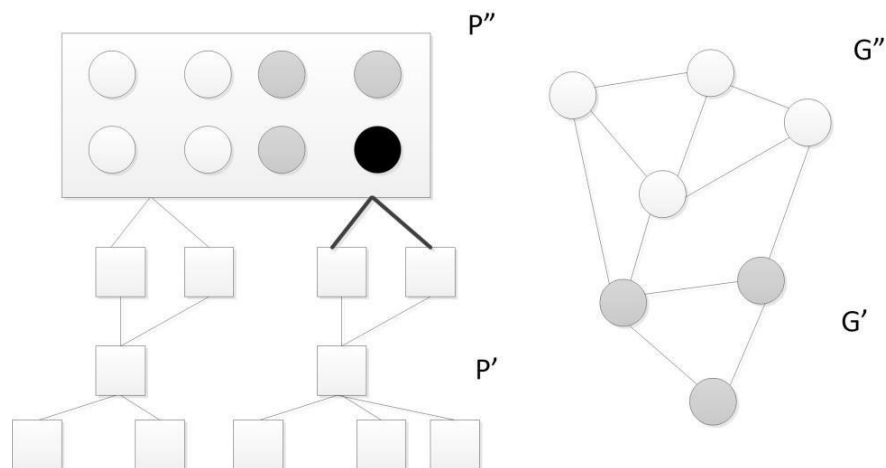


Рисунок 3.2 – Вибраний лідер розсилає запити в межах ЛГП

Крок 3. Якщо отримано позитивну відповідь, то дана множина вузлів фіксується, також встановлюються межі пошукового простору.

Потім вузол-лідер моделює розміщення обчислювальних підзадач по вибраних вузлах, вирішуючи таким чином оптимізаційну завдання розміщення навантаження.

Крок 4. Якщо отримано негативну відповідь у випадку, якщо в аналізованій ЛГП немає вузлів з достатніми для розміщення ресурсами, то ЛГП розширюється таким чином: кожен вузол ЛГП перенаправляє запит від «початкового» вузла далі по ЛГП (рисунок 3.3).

Ця процедура є ітераційною і повторюється доти, поки вузли з необхідними ресурсами не будуть знайдені.

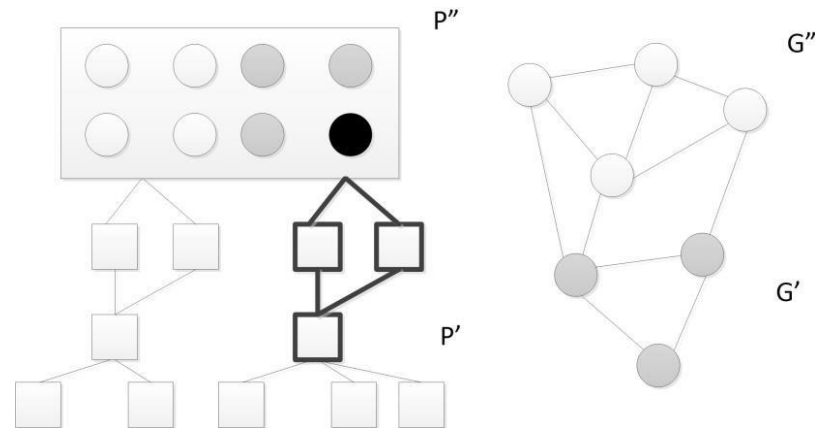


Рисунок 3.3 – Процедура поширення запитів в межах ЛГП

Крок 5. Якщо розміщення навантаження вирішене, і результат задовольняє всім обмеженням завдання Інтернету речей, то обчислювальне підзавдання зв'язується з фіксованою множиною вузлів. Якщо прийнятне рішення не отримано, процедура повторюється, починаючи з кроку 4 (розширення ЛГП). Схема даного методу представлена на рисунку 3.4.

Для часової оцінки даного методу, розглянемо такі параметри:

- N – кількість вузлів, на якому від самого початку виконується обчислювальне завдання;

- N_t – кількість завдань, що підлягають розміщенню;

- L_i – розмір ЛГП для i -го запиту;

- I_g – загальна кількість ітерацій;

- S – кількість ітерацій, необхідна для формування ЛГП з можливістю вирішити завдання.

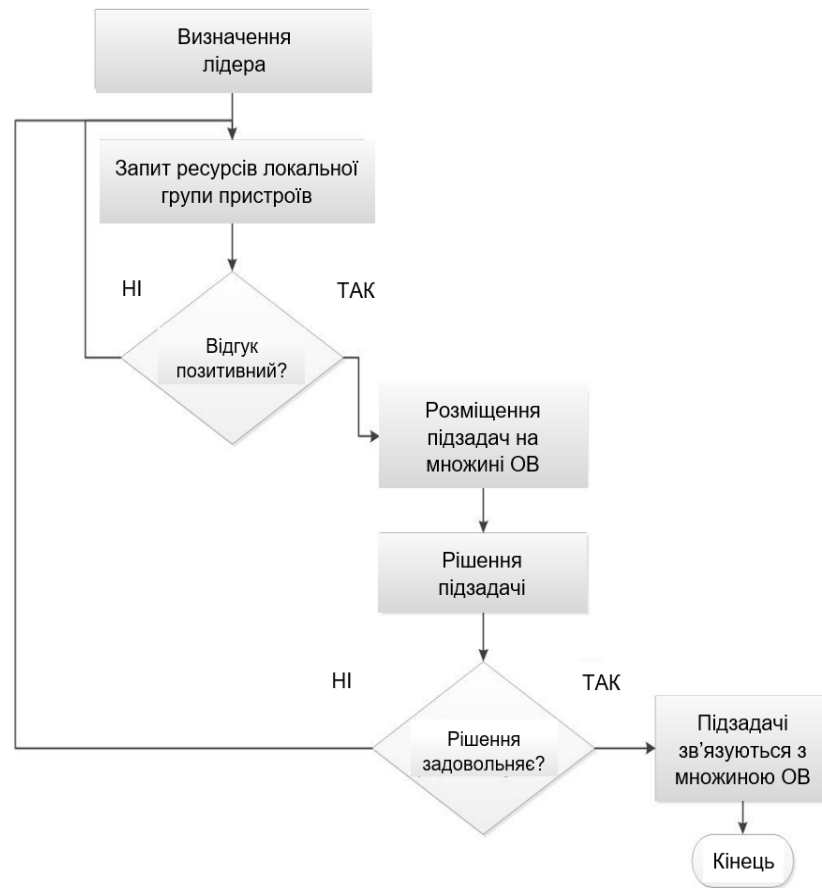


Рисунок 3.4 – Схема методу рішення ЗПЗН з використанням ЛГП

Виходячи з наведених вище параметрів, наближена часова оцінка методу, заснованого на формуванні ЛГП має вигляд:

$$T_{LDG} = \alpha N + \left[\sum_{i=1}^s (vL_i) + N_t L_s \gamma \right] I_g, \quad (3.1)$$

де α – коефіцієнт, що зв'язує час вибору вузла-лідера і кількість вузлів;
 v – коефіцієнт, що зв'язує час обробки запитів та кількість вузлів, які мають бути опитані;
 γ – коефіцієнт, що зв'язує розмір завдання розподілу навантаження туманного шару і час його виконання.

Якщо розглядати метод на основі ЛГП стосовно рішення завдання розподілу навантаження туманного шару, то його можна вважати універсальним, оскільки цей метод не залежить від структур паралельних алгоритмів, використовуваних для рішення завдань Інтернету речей. Однак

також слід відзначити наступне. Вузол, що виробляє спроби розподілу навантаження, в початковий момент часу не має ніяких відомостей щодо стану ресурсів вузлів, що належать локальній групі. Таким чином, пошук варіантів розміщення проводиться «наосліп» і зводиться до методу спроб і помилок. Таким чином, ітераційний характер методу на основі ЛГП зумовлює неможливість передбачити час завершення його роботи.

«Найгірший» варіант роботи методу на основі ЛГП полягає в тому, що вузлу-лідеру доведеться опитати всі вузли-кандидати, розташовані в туманному шарі. Це, по-перше, навряд чи можна буде реалізувати, а по-друге, може зайняти невизначений час.

Як було зазначено вище, часові витрати, пов'язані, в даному випадку, з рішенням завдання реконфігурації системи, здійснюють негативний вплив на ефективність функціонування розподіленої системи, зокрема, туманного середовища. Цей ефект посилюється тим фактом, що в туманному середовищі, яке є динамічним і нестабільним, завдання реконфігурації може розв'язуватися багаторазово. При застосуванні методу ЛГП пошук варіанта розміщення може зайняти невизначений час, що, в кінцевому результаті негативно позначиться на ефективності функціонування системи.

Отже, для досягнення мети, яка полягає у мінімізації часу рішення завдання розподілу навантаження туманного шару, необхідно вирішити такі часткові завдання:

- по-перше, для перенесення обчислювального навантаження необхідно мати інформацією про множину доступних обчислювальних вузлів;

- по-друге, дуже бажано, щоб ця множина не була надмірною, оскільки в умовах перебору варіантів рішень збільшення простору пошуку веде або до погіршення якості рішення, або збільшення часу його здобуття.

Таким чином, актуалізується проблема, пов'язана з розробкою методів та алгоритмів вирішення завдання розподілу навантаження туманного шару, що використовується для виконання запитів Інтернету речей.

3.2 Методика розміщення навантаження в туманному середовищі

Методику розміщення навантаження в середовищі туманних обчислень розглянемо у вигляді блок-схеми, наведеної на рисунку 3.5.

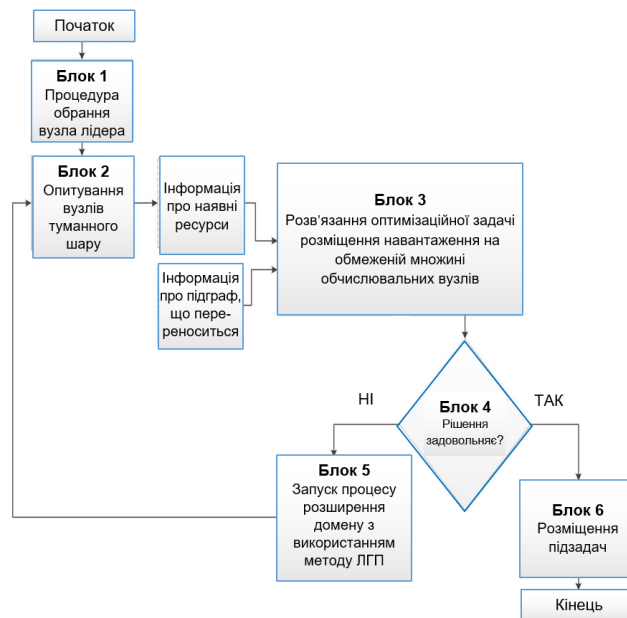


Рисунок 3.5 – Блок-схема методики розміщення навантаження в середовищі туманних обчислень

Блок 1 – Процедура обрання вузла-лідера. Цей блок є необхідним етапом для здійснення збору інформації про наявні ресурси вузлів, які потенційно готові до розміщення обчислювального навантаження.

Умовимося, що у якості вузла-лідера, відповідального за перенесення навантаження, вибирається вузол із множини вузлів, пов'язаних з виконанням обчислювального підзавдання, призначеного до перенесення. При цьому вибір лідера може здійснюватися будь-яким алгоритмом, що використовується для рішення завдання щодо вибору лідера.

Блок 2 – Методика вибору стратегії опитування вузлів туманного шару. Цей блок призначений для виконання обраним вузлом лідером таких етапів:

- опитування вузлів, що належать одному домену;
- збір інформації про наявні ресурси даних вузлів (відповідно до блоку припущень: кожний вузол може надавати інформацію про наявні ресурси).

Інформація про наявні ресурси містить такі дані про вузли:

- завантаженість вузла;
- продуктивність вузла;
- віддаленість вузла від вузла-лідера.

Проведення опитування починається з вузла-лідера і потім поширюється по всьому домену. Важливо відзначити, що дана процедура може бути здійснена з використанням різних стратегій опитування, що мають у своїй основі хвильові алгоритми пошуку як у глибину і ширину. У зв'язку з цим у наступному підрозділі проведено дослідження різних стратегій пошуку стосовно до розв'язуваної задачі і запропоновано методику вибору стратегії опитування вузлів туманного шару.

Інформація про підграф, що переноситься, представляє собою вхідну інформацію і містить такі дані:

- клас паралельного алгоритму;
- модель розпаралелювання;
- спосіб розбиття графа обчислювальної завдання;
- вихідне розміщення (туман, хмара, кінцеві пристрої);
- регламентоване час виконання алгоритму;
- параметри моделей (наприклад, тривалість сезону для острівний моделі паралелізму);
- вимоги до ресурсів.

Блок 3 – Рішення оптимізаційної завдання розміщення навантаження на обмеженій множині обчислювальних вузлів. Цей блок здійснює моделювання розміщення на обмеженій множині вузлів-кандидатів.

Блок 4 – це блок вибору: рішення задовольняє чи ні? У даному блоці відбувається вибір наступного кроку. При позитивній відповіді здійснюється перехід до блоку 6, що відповідає за розміщення підзадач на обмеженій множині обчислювальних вузлів, результат якого полягає в зв'язуванні переносимих підзадач з вузлами-кандидатами.

При негативній відповіді здійснюється перехід до блоку 6, що запускає процес розширення домену з використанням методу ЛГП, котрий дозволяє розширювати від самого початку аналізований домен з використанням локальних груп пристроїв, принцип дії яких докладно описаний в попередньому підрозділі. Після першої ітерації розширення домену, множина вузлів, що входять в оновлений домен, надходить до блоку 2 та етапи, відображені в даній методиці (блок 2 - блок 3 - блок 4 - блок 5) повторюються до отримання позитивної відповіді.

3.3 Вибір стратегії опитування вузлів туманного шару

При виборі стратегії опитування вузлів туманного шару будемо вважати, що, по-перше, існують періоди часу, протягом яких мережа має фіксовану топологію, по-друге, канали зв'язку, що розглядаються, є двоспрямованими. Будемо також вважати, що опитування вузлів туманного шару реалізується за допомогою хвильових алгоритмів.

Хвильові алгоритми – це алгоритми пошуку шляху, які використовують хвильове поширення визначення найкоротшого шляху від початкової вершини до цільової вершини. Назва алгоритму дана не випадково, поведінка алгоритму відповідає поширенню хвилі. Ці алгоритми мають неочевидну, вкрай корисну для розподілених застосувань властивість: вони дуже легко реалізуються для розподіленого випадку (коли кожний обчислювальний вузол кожного моменту має власні поточні характеристики, немає "спільної шини" і "загального годинника" – всі спілкуються тільки зі своїми сусідами і не володіють загальною картиною). Хвиля огинає перешкоди, поступово заповнюючи весь простір. Хвильові алгоритми відрізняються великою різноманітністю своїх властивостей. Серед основних аспектів, що дозволяють диференціювати хвильові алгоритми виділяють такі:

- централізацію (децентралізацію);
- топологію (кільце, дерево, кліки і т.п.);

- початкові відомості (самоідентифікація, ідентифікація сусідів, сприйняття напряму);
- число рішень;
- складність (за числу обмінів повідомленнями, обчислювальна складність алгоритму).

Оцінка обчислювальної складності алгоритмів за числом обмінів повідомленнями (O_1) і обчислювальною складністю алгоритмів за часом (O_2) наведена в таблиці 3.1.

Таблиця 3.1 – Оцінка складності хвильових алгоритмів

Алгоритм	Топологія	Централізація	O_1	O_2
Загальні алгоритми				
Кільцевий	Кільце	Централізований	N	N
Дерев'яний	Дерево	Децентралізований	N	$O(D)$
Відлуння	Довільна	Централізований	$2 E $	$O(N)$
Опитування	Кліка	Централізований	$2N-2$	$2N-2$
Фазовий	Довільна	Децентралізований	$2D E $	$2D$
Фазовий на кліках	Кліка	Децентралізований	$N(N-1)$	2
Фіна	Довільна	Децентралізований	$\leq 4N E $	$O(D)$
Алгоритми обходу				
Послідовне опитування	Кліка	Централізований	$2N-2$	$2N-2$
Тора	Тор	Централізований	N	N
Гіперкуба	Гіперкуб	Централізований	N	N
Таррі	Довільна	Централізований	$2 E $	$2 E $
Алгоритми пошуку в глибину				
Класичний	Довільна	Централізований	$2 E $	$2 E $
Авербаха	Довільна	Централізований	$4 E $	$4N-2$
Сідона	Довільна	Централізований	$4 E $	$2N-2$
З урахуванням сусідів	Довільна	Централізований	$2N-2$	$2N-2$

У таблиці 3.1 N позначає кількість процесів, $|E|$ – кількість каналів зв'язку, а D – діаметр мережі (вимірюваний по числу переходів). Крім того, топологія «кліка» (згідно теорії графів) визначає підмножини попарно суміжних вершин неорієнтованого графа.

З метою вибору алгоритму, котрий найкращим чином підходить до рішення поставленого завдання, тобто. реалізації процедури опитування кінцевого числа вузлів туманного шару, пропонується наступний підхід. Він базується на поінформованості процесів про топологію мережі і відомостях щодо локалізації мережі в туманному шарі.

Під обізнаністю процесів будемо розуміти ту інформацію про розподілену систему, яку надано в початковому стані процесів. До таких відомостей відносять таку інформацію:

- топологічна інформація;
- відомості про кількість процесів;
- діаметр графа мережі;
- топологія мережі;
- відмінні ознаки процесів (ідентифікація процесів);
- відмінні ознаки сусідів, тобто кожен процес знає ідентифікатори своїх сусідів.

Особливості туманних середовищ накладають обмеження на вибір стратегії опитування вузлів туманного шару, зокрема, локалізація вузлів-кандидатів для перенесення обчислювального навантаження в межах туманного шару грає важливу роль в виборі алгоритму опитування.

Наприклад, вузли-кандидати для перенесення обчислювального навантаження можуть локалізуватися в тій частині туманного шару, де розташовуються вузли, виступаючи в ролі агрегують пристроїв (шлюзи, мікроконтролери).

Для подальшої передачі агрегованої інформації в хмару необхідно використовувати високонадійні канали зв'язку з широкою смугою

пропускання (електричний кабель або радіоканал із використанням систем широкопasmового доступу).

Таким чином, аналізована ділянка туманного шару повинна бути статичною і забезпеченою високонадійними, широкопasmовими каналами зв'язку.

У свою чергу, ділянка мережі, де розташовано велику кількість сенсорних пристроїв, причому їх число постійно змінюється, характеризується більшою динамічністю.

Взаємодія між сенсорними пристроями і шлюзами відбувається на основі бездротових зв'язків, таких як Bluetooth і LoRA, мобільного зв'язку поколінь 4G і 5G, протоколів ZigBee, Wi-Fi . На даний момент характеристики аналізованих комунікаційних каналів поступаються по надійності та швидкості передачі характеристикам провідних зв'язків між шлюзами та серверами.

Для вибору алгоритму опитування вузлів туманного шару необхідно прийняти такі припущення:

- чим ближче вузли-кандидати розташовані до хмари, тим топологія мережі більш статична, показники каналів зв'язку кращі, тобто надійніше, ширше смуга пропускання тощо;

- чим вузли-кандидати розташовані ближче до краю мережі, тим топологія мережі більше динамічна, а характеристики каналів зв'язку гірше.

Дані залежності наочно продемонстровані на схемі, наведеної на рисунку 3.5.

Перейдемо до опису процесу вибору стратегії опитування вузлів туманного шару, спираючись на властивість поінформованості процесів і прийняті вище припущення.

Такі аспекти поінформованості як самоідентифікація та ідентифікація сусідів будемо вважати відомими за замовчуванням. Також будемо вважати, що вузол-лідер має уявлення щодо орієнтації мережі, тобто початкова топологія мережі відома.

Якщо вузол-лідер має уявлення щодо орієнтації мережі, то йому відомі відомості про топологічну інформацію графа мережі:

- кількість процесів;
- діаметр графа мережі;
- топологія мережі.

У цьому випадку можна провести аналіз віддаленості вузла-лідера від хмари.

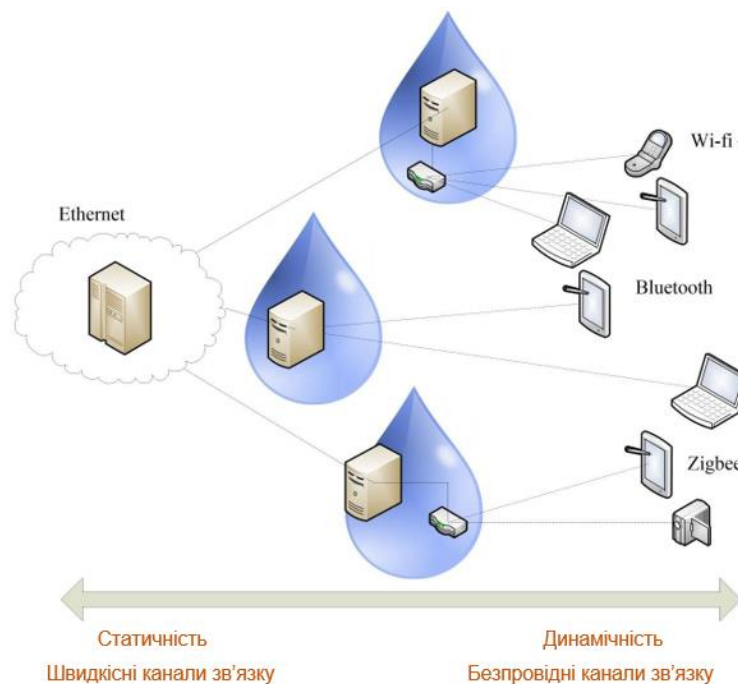


Рисунок 3.5 – Схема залежності статичності та характеристик каналів зв'язку від локалізації в туманному арі

Результатом цього аналізу є значення параметрів D_1 та D_2 , що вимірюються в хопх (hop – транзитна ділянка). При цьому D_1 – відстань від вузла-лідера до «хмари», D_2 – відстань від вузла-лідера до «краю» мережі.

На підставі оцінки значень параметрів D_1 і D_2 та прийнятих вище припущень відбувається прийняття рішення про те, які критерії повинні бути враховані під час виборів алгоритму опитування. Далі на підставі наявних оцінок складності хвильових алгоритмів, зробити висновок про ефективність застосування того або іншого алгоритму опитування в конкретному випадку.

Наприклад, при розташування вузла-лідера ближче до хмари, тобто, коли $D1 < D2$, обчислювальна складність алгоритму за числом повідомлень не грає першорядної ролі при виборі алгоритму опитування, оскільки вузли в даній частині мережі пов'язані між собою високошвидкісними каналами зв'язку і мають велику обчислювальну потужність, що дозволяє проводити процедуру опитування без оврахування затримок на передачу даних мережею. Ethernet є широко поширеною і домінуючою технологією для провідних локальних мереж, швидкість систем Ethernet досягає 100 Гбіт/с, наприклад, оптоволоконний кабель забезпечує швидкість передачі даних до 1 Гбіт/с, бездротова телекомунікаційна технологія WiMax також забезпечує швидкість до 1 Гбіт/с. У цьому випадку вибір алгоритму опитування здійснюється на підставі знань про топологію мережі, тобто. з числа наявних хвильових алгоритмів вибирається алгоритм з мінімальною обчислювальною складністю для аналізованої топології.

Якщо ж вузли-кандидати для переміщення обчислювального навантаження розташовуються ближче до «краю» мережі, тобто. $D1 > D2$, то вибір алгоритму опитування відбувається на підставі комплексної оцінки складності хвильового алгоритму, включає складність по обміну повідомленнями і складність за часом.

Це пояснюється тим, що швидкість передачі даних в бездротових мережах, використовуваних для реалізації технології Інтернету речей, значно нижче швидкості, яку забезпечують мережні технології, що реалізують зв'язок з хмарними центрами зберігання та обробки даних. Більше того, як правило, вузли туманного шару мають низькі обчислювальні ресурси. Наприклад, максимальна швидкість передачі в безпроводній мережі Zigbee складає 256 Кбіт/с, а реальна швидкість варіюється в інтервалі від 5 до 40 Кбіт/с.

Крім врахування комплексної оцінки складності при виборі алгоритму опитування, також варто враховувати топологію мережі. Розглянемо послідовність дій при виборі алгоритму опитування на основі первісних

відомостей про процеси та інформацію про локалізації вузлів-кандидатів у туманному шарі.

Як перший «кращий» алгоритм зі списку алгоритмів вибирається алгоритм, готрий задовольняє вимогам щодо топології мережі. Далі послідовно перебираються і аналізуються алгоритми, що перебувають в списку, до останнього.

Проведемо теоретичну оцінку виграшу за часом при виборі алгоритму опитування вузлів туманного шару в випадку розташування вузла-лідера ближче до хмарного шару.

При оцінці обчислювальної складності за часом фазового алгоритму та алгоритму з урахуванням сусідів при лінійній топології обчислювальної мережі, виграш за часом відсутній, оскільки відношення часових оцінок даних алгоритмів дорівнює одиниці.

При оцінці обчислювальної складності за часом фазового алгоритму (O_ϕ) і алгоритму з урахуванням сусідів (O_c) при топології обчислювальної мережі кільце, виграш за часом досягає приблизно двох раз, оскільки

$$\frac{O_\phi}{O_c} = \frac{2D}{2N-1} = \frac{2 \left\lceil \frac{N}{2} \right\rceil}{2(N-1)} \approx \frac{1}{2}, \quad (3.2)$$

де при топології кільце $D = \left\lceil \frac{N}{2} \right\rceil$, причому $[N]$ – ціла частина числа N .

При оцінці обчислювальної складності за часом фазового алгоритму (O_ϕ) і алгоритму з урахуванням сусідів (O_c) при топології обчислювальної мережі 2D-решітка отримаємо

$$\frac{O_\phi}{O_c} = \frac{2D}{2N-1} = \frac{2 \times 2(\sqrt{N}-1)}{2(N-1)} = \frac{2}{\sqrt{N}+1}. \quad (3.3)$$

Тоді, якщо, наприклад, $N = 100$, то виграш за часом при виборі фазового алгоритму складає приблизно 5,5 разів. При оцінці обчислювальної складності за часом фазового алгоритму (O_ϕ) і алгоритму з урахуванням сусідів (O_c) при топології обчислювальної мережі гіперкуб, діаметр (D) обчислювальної мережі визначається формулою $\log_2 N$. Тоді, якщо, наприклад, $N = 100$, то виграш за часом при виборі фазового алгоритму досягає 15 разів.

Зазначимо, що для конфігурацій з більшою кількістю вузлів, виграш за часом буде збільшуватись.

3.4 Оцінка ефективності методу розподілу навантаження туманного шару Інтернету речей

Для проведення оцінки ефективності запропонованого методу будемо припускати, що завдання, що розв'язується, має рішення. Тоді оцінка часу для реалізації методу розподілу навантаження туманного шару Інтернету речей має наступний вигляд:

$$T_M = \alpha N + kF + \gamma N_t F_s, \quad (3.4)$$

де N – кількість вузлів, на яких від самого початку виконувалася обчислювальна завдання; αN – час, необхідний для проведення процедури вибору лідера; F – кількість вузлів в туманному шарі; kF – час, необхідний для опитування вузлів з використанням алгоритмів опитування; F_s – кількість вузлів після процедури виключення непотрібних вузлів; $\gamma N_t F_s$ – час, необхідний для процедури моделювання розміщення завдань N_t по вузлах F_s .

Розглянемо кожний доданок формули (3.4) детальніше.

Час, необхідний для проведення процедури вибору лідера (αN),

визначається часовою складністю алгоритму, що використовується для цієї мети.

Розглянемо різні випадки поінформованості вузла-лідера про топологію мережі. Випадок, коли топологія мережі відома. Якщо топологія мережі відома і має вигляд дерева, то обрання лідера можна здійснити, скориставшись деревним алгоритмом. Тоді на рішення завдання про вибіравитрачається $O(D)$ одиниць часу з використанням $O(N)$ обмінів повідомленнями (табл. 3.1). За допомогою фазового алгоритму обрання лідера можна провести в мережах з довільною топологією, використовуючи $O(D|E|)$ обмінів повідомленнями та витрачаючи $O(D)$ одиниць часу. При рішенні завдант з вибором алгоритму Фіна складність за числом обмінів повідомленнями становить $O(N^2|E|)$ (табл. 3.1). Мережі з невідомою топологією, але сприйняттям напряду зменшується складність проведення виборів лідера з величини $O(M\log N + m)$ до величини $O(M\log N)$. Складність за числом обмінів повідомленнями алгоритму GHS (алгоритм Галладжера - Гамблета – Спіри) складає $O(|E| + M\log N)$, причому часова складність може бути знижена до $O(N)$.

Величина kF – час, необхідний для опитування вузлів із використанням алгоритмів опитування, залежить від вибору стратегії опитування, яка, у свою чергу визначається методикою вибору, описаною у попередньому підрозділі. У цьому випадку також необхідно розглянути випадки про наявність чи відсутність поінформованості про топології мережі. Якщо топологія мережі відома, наприклад, топологія мережі – гіперкуб, тоді обчислювальна складність алгоритму обходу за часом становитиме N , де N – кількість вузлів у туманному шарі. Якщо топологія мережі невідома, тоді розглядаємо алгоритми опитування, працюючі на довільній топології мережі. Наприклад, обчислювальна складність за часом алгоритму Сідону становить $2N - 2$, де N – кількість вузлів в туманному шарі.

Час, необхідний для процедури моделювання розміщення завдань N_t по вузлах F_s . визначається відповідним оптимізаційним алгоритмом, таким

як, наприклад, випадковий пошук, метод гілок та границь, мурашині алгоритми, генетичний алгоритм тощо.

Проведемо дослідження залежності часу, необхідного на реалізацію запропонованого методу, математична модель якого описана формулою (3.4), від різних параметрів, розглянутих вище.

3.4.1 Залежність часу реалізації розробленого методу від вибору алгоритму обрання лідера

Розглянемо випадки.

A1 – обрання лідера за допомогою алгоритму Фіна на ієрархічній мережі.

A2 – обрання лідера за допомогою GHS алгоритму на довільній мережі.

A3 – обрання з лідера за допомогою ієрархічного алгоритму.

Проведемо дослідження оцінки за часом вищеперелічених алгоритмів в залежності від значення кількості задіяних вузлів – параметра N (рисунок 3.6).

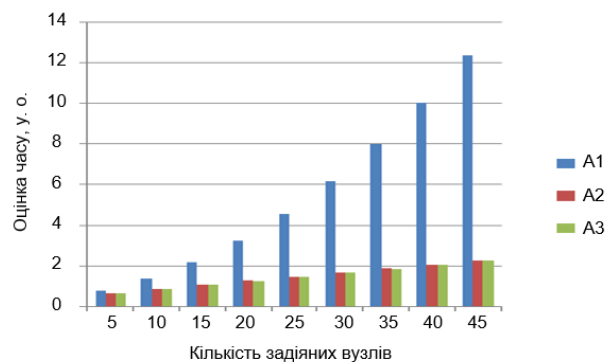


Рисунок 3.5 – Залежність часової оцінки розробленого методу при використанні різних алгоритмів обрання лідера від кількості задіяних вузлів

З рисунка 3.5 бачимо, що час, який витрачається алгоритмом Фінна, різко зростає із збільшенням параметра N , а час, що витрачається ієрархічним і GHS алгоритмами приблизно однаково і має незначне зростання із збільшенням N .

Тепер проведемо порівняння часу, необхідного для реалізації розробленого методу при різних варіаціях алгоритмів обрання лідера з часом, необхідним для реалізації методу формування обмежень з використанням стандартного алгоритму (A4) при великій кількості ітерацій (рисунок 3.6).

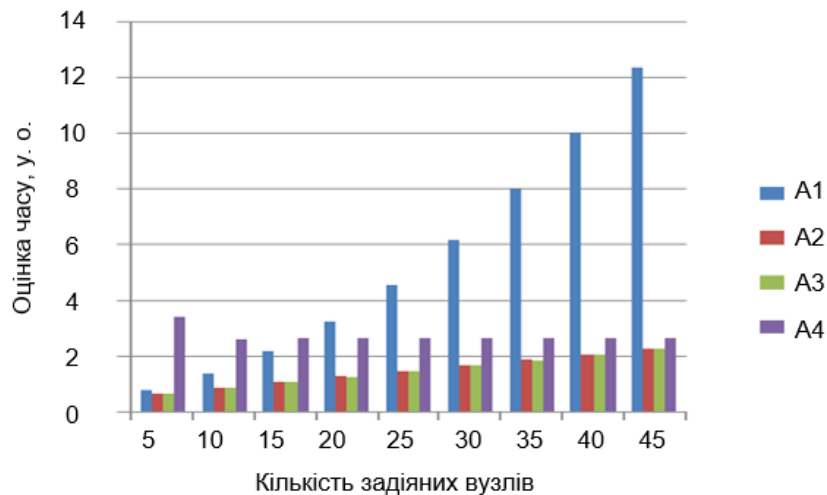


Рисунок 3.6 – Часова оцінка розробленого методу та методу на основі локальних груп пристроїв

З рисунка 3.6 бачимо, що стандартний метод в меншій ступеня залежить від кількості вузлів, на яких від самого початку виконувалася завдання. При цьому час, що витрачається методом на основі локальних груп пристроїв вище, чим час, що витрачається розробленим методом, особливо при невеликій кількості пристроїв. Зауважимо, що у туманному шарі зазвичай розглядаються невеликі групи туманних пристроїв.

Отже, можна говорити про те, що ефективність розробленого методу перевищує ефективність аналога, причому найбільш суттєвим перевищення є при кількості задіяних пристроїв, що не перевищує 30.

Таким чином, можна зробити висновок про те, що запропонований метод з використанням GHS алгоритму обрання лідера мало залежить від кількості вузлів, на яких спочатку виконувалося завдання та перевищує ефективність свого аналога до двох з половиною разів.

3.4.2 Оцінка часу моделювання розподілу навантаження

По-перше будемо змінювати число ітерацій пошукових алгоритмів. Промодельюємо кількість можливих ітерацій в інтервалі [1; 45] (рисунок 3.7).

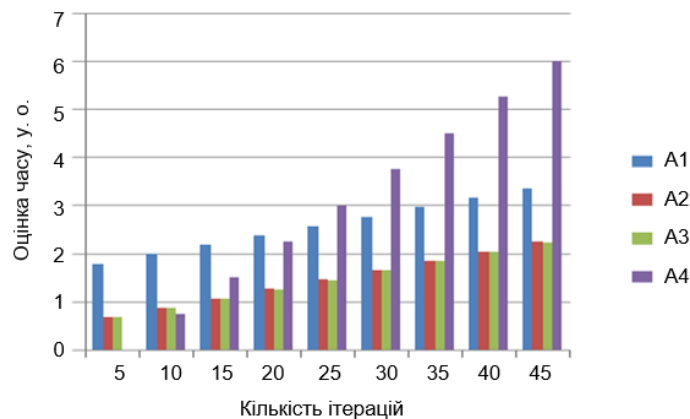


Рисунок 3.7 – Часові оцінки запропонованого методу та стандартного методу при зміні числа ітерацій у інтервалі [1; 45]

З рисунка 3.7 бачимо, що час, що витрачається стандартним методом, до двох раз більше, чим час, що витрачається розробленим методом. Причому, величина необхідного для реалізації часу різко зростає з збільшенням числа ітерацій, як можна побачити на рисунку 3.8.

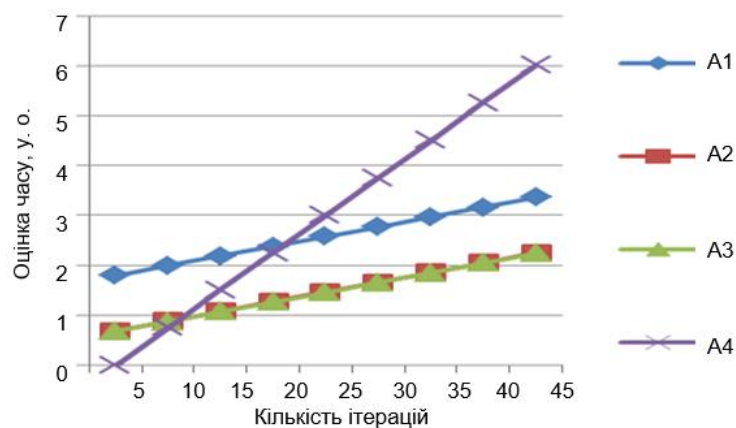


Рисунок 3.8 – Аналіз часових оцінок запропонованого методу та стандартного методу при зміні числа ітерацій у інтервалі [1; 45]

На рисунку 3.8 бачимо область значень, у якій часова оцінка розробленого методу та методу на основі ЛГП однакові. Число ітерацій при цьому лежить в інтервалі [10; 20].

Таким чином, можна зробити висновок про те, що запропонований метод ефективно використовувати, якщо кількість ітерацій у методі-аналогі перевищує десять ітерацій.

Тепер досліджуємо зміни часу, необхідного на реалізацію пропонованого методу, в залежності від вибору алгоритму опитування вузлів туманного шару (на початковому етапі методу одноразовому опитуванню підлягає доволі велика кількість вузлів туманного шару):

V1. Ієрархічний алгоритм на топології мережі «дерево».

V2. Алгоритм Фіна на довільній топології мережі.

V3. Алгоритм з урахуванням сусідів на довільній топології мережі.

Для дослідження будемо варіювати параметр кількості вузлів туманного шару F (рисунок 3.9).

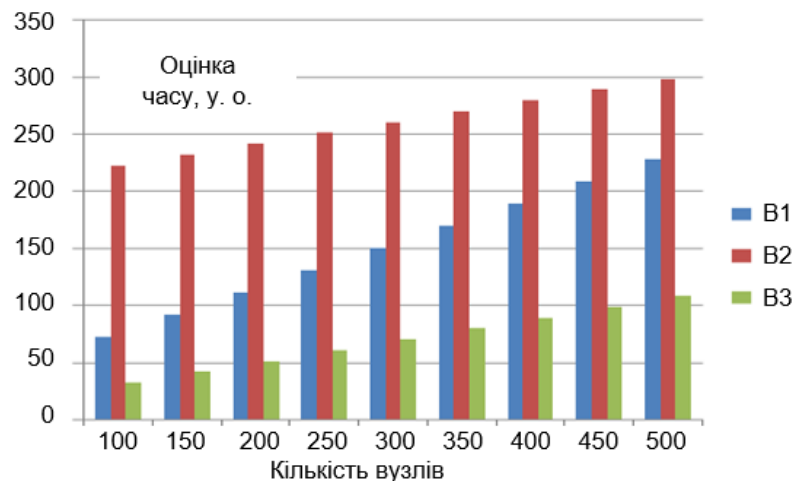


Рисунок 3.9 – Часові оцінки запропонованого методу під час використання різних варіантів алгоритмів опитування при зміні параметра F

З рисунка 3.9 бачимо, що найефективнішим алгоритмом опитування вузлів туманного шару є ієрархічний алгоритм, орієнтований на фіксовану топологію мережі «дерево». Якщо говорити про довільну топологію мережі,

то більш ефективним є алгоритм опитування з урахуванням сусідів. Алгоритм Фінна на довільній топології мережі вимагає найбільшу кількість часу на свою реалізацію і тому в даному випадку він є неефективним. Виходячи з цього, його недоцільно розглядати далі в порівнянні зі стандартним методом опитування вузлів туманного шару.

Проведемо порівняння необхідного часу для загального опитування вузлів туманного шару. Для цього розглянемо три таких варіанти:

C1. Ієрархічний алгоритм, що базується на сталій топології мережі «дерево».

C2. Алгоритм з урахуванням сусідів, що базується на довільній топології мережі

C3. Стандартний алгоритм загального опитування вузлів туманного шару, що суттєво не залежить від зміни параметра F .

Для дослідження також будемо варіювати параметр кількості вузлів туманного шару F (рисунок 3.10).

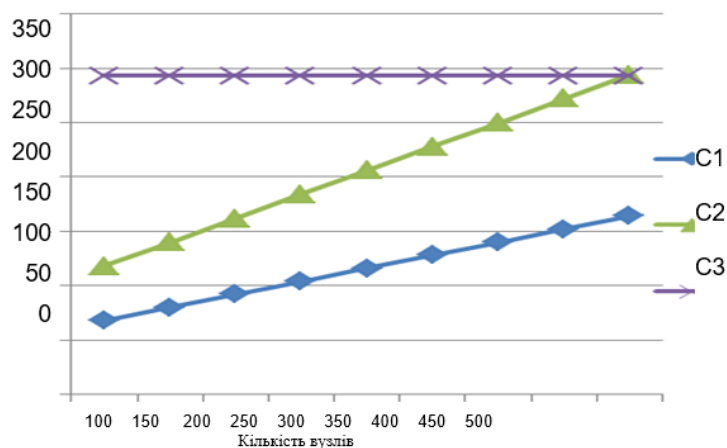


Рисунок 3.10 – Часові оцінки запропонованого методу та стандартного методу при використанні різних варіантів алгоритмів опитування та при зміні параметра F

З рисунка 3.10 бачимо, що стандартний метод менше ефективний в порівнянні із запропонованими методами.

Запропонований підхід до формування локальної групи пристроїв з метою надання фізичної бази для розподілу навантаження у туманному середовищі. Підхід полягає в формуванні обмежень на множину обчислювальних пристроїв в режимі реального часу. Обирається вузол-лідер, який ітеративно моделює розміщення обчислювальних підзадач по вибраних вузлах, вирішуючи таким чином оптимізаційну завдання розміщення навантаження. Показано, що з огляду великої кількості вузлів- кандидатів, розміщених у туманному середовищі, процедура вирішення завдання перенесення обчислювальної навантаження може зайняти невизначений час, що, відповідно, негативно позначиться на ефективності функціонування системи. На даній підставі зроблено висновок про необхідність обмеження множини вузлів-кандидатів для розподілу поточних завдань.

Розроблено методику розміщення навантаження в туманному середовищі, що використовує формування обмежень в задачі перенесення обчислювального навантаження. Запропонована методика дозволяє зменшити часові витрати при організації туманних обчислень за рахунок скорочення пошукового простору при перенесення обчислювального навантаження.

Розроблено підхід до вибору стратегії опитування вузлів туманного шару, що дозволяє суттєво скоротити час опитування вузлів туманного шару в залежності від топології мережі за рахунок підбору найбільш ефективних алгоритмів, необхідних для його реалізації.

Проведено аналітичну часову оцінку запропонованого методу, що дозволяє оцінити його ефективність в залежності від використовуваних алгоритмів та порівняно зі стандартним методом без попереднього скорочення множини обчислювальних вузлів туманного середовища.

ВИСНОВКИ

Сукупність отриманих у кваліфікаційній роботі результатів дозволило вирішити актуальне науково-технічне завдання розробки методу розподілу навантаження туманного шару Інтернету речей.

В результаті проведених досліджень отримані такі наукові та практичні результати:

1 Проведений аналіз сучасних тенденцій в області організації розподілених обчислень. Розглянуто технології ґрид-обчислень і хмарних обчислень, наведено приклади їх практичної реалізації в розподілених системах різного призначення. Виявлені проблеми, пов'язані з передачею великих обсягів даних та часом відгуку системи при передачі великих обсягів даних. Розглянуто технологію туманних обчислень, наведено приклади її використання в розподілених системах моніторингу і управління. Позначено головну особливість даної технології, що полягає в можливості зміщення обробки даних суттєво ближче до місця їх генерації, що робить можливим скоротити навантаження на канали зв'язку і знизити латентність системи. Виявлено проблему при реалізації географічно розподілених систем на базі технологій туманних обчислень, яка полягає в наявності невиправданих суттєвих часових витрат при організації обчислювального процесу, що пов'язані з багаторазовим виконанням завдання перенесення обчислювального навантаження внаслідок динамічності туманного середовища. На основі проведеного аналізу зроблено висновок, що поставлене в роботі завдання щодо методу розподілу навантаження туманного шару Інтернету речей, є актуальним, а його вирішення дозволить більш ефективно використовувати ресурси туманного середовища.

2 Розглянуто питання перенесення навантаження до туманного шару в системах Інтернету речей. Описано моделі розпаралелювання завдань Інтернету речей з використанням обчислювальних вузлів туманного шару. Варіанти реалізації моделей розглянуті на прикладі розвантаження хмарного

середовища шляхом паралельного використання туманних вузлів для деяких складових розглядаємого завдання. Проведено аналіз моделей і методів рішення завдання перенесення обчислювальної навантаження хмарного і туманного середовища. Виявлено, що більшість відомих моделей та методів не враховують наявність транзиторних ділянок, що з огляду великої географічної розподіленості систем, функціонуючих в туманному середовищі, є важливим фактором, що обмежує їх використання. Сформульована постановка завдання перенесення навантаження до туманного шару в системах Інтернету речей та розроблена математична модель розподілу відповідного навантаження.

3 Розроблений метод розподілу навантаження туманного шару Інтернету речей. Запропонований підхід до формування локальної групи пристроїв з метою надання фізичної бази для розподілу навантаження у туманному середовищі. Підхід полягає в формуванні обмежень на множину обчислювальних пристроїв в режимі реального часу. Показано, що з огляду великої кількості вузлів- кандидатів, розміщених у туманному середовищі, процедура вирішення завдання перенесення обчислювальної навантаження може зайняти невизначений час, що, відповідно, негативно позначиться на ефективності функціонування системи. На даній підставі зроблено висновок про необхідність обмеження множини вузлів-кандидатів для розподілу поточних завдань. Розроблено методику розміщення навантаження в туманному середовищі, що використовує формування обмежень в задачі перенесення обчислювального навантаження. Запропонована методика дозволяє зменшити часові витрати при організації туманних обчислень за рахунок скорочення пошукового простору при перенесення обчислювального навантаження. Розроблено підхід до вибору стратегії опитування вузлів туманного шару, що дозволяє суттєво скоротити час опитування вузлів туманного шару в залежності від топології мережі за рахунок підбору найбільш ефективних алгоритмів, необхідних для його реалізації.

4 Досліджена ефективність методу розподілу навантаження туманного шару Інтернету речей. Зокрема, проведено аналітичну часову оцінку запропонованого методу, що дозволила оцінити його ефективність в залежності від використовуваних алгоритмів та порівняно зі стандартним методом без попереднього скорочення множини обчислювальних вузлів туманного середовища.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1) Z. Kolesnyk, O. Mezhenskyi, O. Davykoza, H. Kuchuk Fog computing technology in distributed systems. Системи управління, навігації та зв'язку. Полтава: Національний університет «Полтавська політехніка імені Юрія Кондратюка», 2024. Вип. 1(75). С. 94–100. doi: 10.26906/SUNZ.2024.1.094.
- 2) Fang, J., Zhao, G., Xu, H., Wu, C., Yu, Z. GRID: Gradient Routing with In-Network Aggregation for Distributed Training. IEEE/ACM Transactions on Networking, 31(5), pp. 2267–2280. DOI: 10.1109/TNET.2023.3244794
- 3) Fatlawi, A., Al Dujaili, M.J. (2023), Integrating the Internet of Things (IoT) and Cloud Computing Challenges and Solutions: A Review. AIP Conference Proceedings, 2977(1), 020067. doi: <http://dx.doi.org/10.1063/5.0181842>
- 4) Almurshed, O., Rana, O., Li, Y., Jayaraman, P.P., Dustdar, S. A Fault-Tolerant Workflow Composition and Deployment Automation IoT Framework in a Multicloud Edge Environment. IEEE Internet Computing. 26(4). P. 45–52. DOI: 10.1109/MIC.2021.3078863
- 5) Liu, L., Chen, H., Xu, Z. (2022). SPMOO: A Multi-Objective Offloading Algorithm for Dependent Tasks in IoT Cloud-Edge-End Collaboration. Information, 13, 75. doi: <https://doi.org/10.3390/info13020075>
- 6) Chalapathi, G.S.S., Chamola, V., Vaish, A., Buyya, R. (2022). Industrial internet of things (Iiot) applications of edge and fog computing: A review and future directions. Advances in Information Security, 83, 293–325. doi: https://doi.org/10.1007/978-3-030-57328-7_12
- 7) Qayyum, T., Trabelsi, Z., Waqar Malik, A., Hayawi, K. (2022). Mobility-aware hierarchical fog computing framework for Industrial Internet of Things. Journal of Cloud Computing, 11(1), 72. doi: <https://doi.org/10.1186/s13677-022-00345-y>
- 8) 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.

9) Gomathi B., Saravana Balaji B., Krishna Kumar V., Abouhawwash, M., Aljahdali S., Masud M. and Kuchuk N. (2022), “Multi-Objective Optimization of Energy Aware Virtual Machine Placement in Cloud Data Center”, *Intelligent Automation and Soft Computing*, Vol. 33(3), pp. 1771–1785, doi: <http://dx.doi.org/10.32604/iasc.2022.024052>

10) Xu L. Da, He W., Li S. Internet of things in industries: A survey. *IEEE Trans. Ind. informatics*. 2014. T. 10. No. 4. P. 2233–2243. DOI: 10.1109/TII.2014.2300753

11) Tang B., Chen Z., Hefferman G., Wei T., He H., Yang Q. A Hierarchical Distributed Fog Computing Architecture for Big Data Analysis in Smart Cities. *ASE BD&SI 2015: Proceedings of the ASE BigData & Social Informatics*. 2015. P. 1–6. DOI: <http://dx.doi.org/10.1145/2818869.2818898>

12) Tang B., Chen Z., Hefferman G., Pei S., Wei T., He H., Yang Q. Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities. *IEEE Transactions on Industrial Informatics*. 2017. 13(5). P. 2140–2150. DOI: 10.1109/TII.2017.2679740

13) Narayanan LK, Sankaranarayanan S. IoT Enabled Smart Water Distribution and Underground Pipe Health Monitoring Architecture for Smart Cities. *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*. 2019. P. 1–7. DOI: 10.1109/I2CT45611.2019.9033593

14) Stamatescu G., Dragana C., Stamatescu I., Ichim L., Popescu D. IoT – Enabled Distributed Data Processing for Precision Agriculture. *27th Mediterranean Conference on Control and Automation (MED)*. 2019. P. 286–291. DOI: 10.1109/MED.2019.8798504

15) Melnik E., Klimenko A., Klimenko V. A recovery technique for the fog-computing-based information and control systems. *Advances in Intelligent Systems and Computing*. 860. P. 216–227. DOI: 10.1007/978-3-030-00184-1_20

16) Chen R.-Y. Fog computing-based intelligent inference performance

evaluation system integrated internet of thing in food cold chain. 12th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD. 2015. Art. no. 7382059. P. 879–886. DOI: 10.1109/FSKD.2015.7382059

17) 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>.

18) 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>