

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 управління навантаженням в розподілених  
системах за допомогою мобільних агентів

(тема)

Виконав:

студент II курсу, групи СПМ-23-2  
Момотов Є.В.  
(прізвище, ініціали)

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

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

Освітня програма Системне програмування  
(повна назва освітньої програми)

Керівник: проф. Можєв О.О.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Коваленко А.А.

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

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

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

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

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

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

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

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студенту \_\_\_\_\_ Момотову Єгору Валерійовичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Метод управління навантаженням в розподілених системах за допомогою мобільних агентів

затверджена наказом по університету від “ 22 ” листопада 2024 р. № 1236 Ст

2. Термін подання студентом роботи до екзаменаційної комісії \_\_\_\_\_ 20 січня 2025 р.

3. Вхідні дані до роботи операційна система –Windows або Linux,  
1Гб оперативної пам'яті ПК процесор на 1ГГц, 100Мб свобідної пам'яті на жорсткому диску

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

- 1) аналіз управління розподіленими системами на основі мультиагентних технологій;
- 2) дослідження механізму поширення інформації в мультиагентної системи;
- 3) мобільні програмні агенти як інструмент управління навантаженням у розподілених у розподілених інфокомунікаційних системах;
- 4) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 13

---

---

---

---

---

---

---

---

---

---

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

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

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	26.11.24-30.11.24	
2	Вибір та обґрунтування методики		
3	Вибір інструментальних засобів	02.12.24-05.12.24	
4	Розробка методу		
5	Проведення експериментів	11.12.24-21.12.24	
6	Оформлення матеріалів кваліфікаційної роботи	23.12.24-03.01.25 04.01.25-07.01.25	
7	Подання кваліфікаційної роботи її попередній захист	08.01.25-11.01.25	
8	Подання кваліфікаційної роботи на рецензування	13.01.25-17.01.25	

Дата видачі завдання 25 листопада 2024 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

проф. Можасєв О.О. \_\_\_\_\_  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 60 с., 20 рис., 1 дод., 20 джерел.

### МОБІЛЬНІ АГЕНТИ, ІНТЕРНЕТ РЕЧЕЙ, ПЛАТФОРМА, ПРОТОКОЛ, МІГРАЦІЯ СЛУЖБ

Метою кваліфікаційної роботи є розробка засобів спеціального математичного та програмного забезпечення процесів управління інфокомунікаційними службами (об'єктами ІС) на основі реалізації систем мобільних агентів.

В ході виконання кваліфікаційної роботи розроблено механізм переміщення мобільних агентів для збереження якості обслуговування або виключення порушення працездатності платформ у разі штатних або нерегламентованих збурень (системні збої, зовнішні атаки, наприклад). У таких ситуаціях мобільні агенти мігруватимуть з повністю або частково втратили працездатність платформ на платформи, що коректно функціонують, для підтримки QoS на необхідному рівні. Розроблено архітектуру системи, визначено процедури міграції агентів та проведено моделювання для запропонованої схеми міграції служб на основі агентів.

## ABSTRACT

Master's thesis: 60 pages, 20 figures, 1 appendices, 20 sources.

### MOBILE AGENTS, INTERNET OF THINGS, PLATFORM, PROTOCOL, MIGRATION OF SERVICES

The purpose of the qualification work is to develop special mathematical and software tools for the management of infocommunication services (IS objects) based on the implementation of mobile agent systems.

During the qualification work, a mechanism for moving mobile agents was developed to maintain the quality of service or eliminate disruption of the platforms in the event of regular or unregulated disturbances (system failures, external attacks, for example). In such situations, mobile agents will migrate from fully or partially inoperable platforms to platforms that are functioning correctly to maintain QoS at the required level. The system architecture was developed, agent migration procedures were defined, and modeling was performed for the proposed agent-based service migration scheme.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	7
ВСТУП .....	8
1 АНАЛІЗ УПРАВЛІННЯ РОЗПОДІЛЕНИМИ СИСТЕМАМИ НА ОСНОВІ МУЛЬТИАГЕНТНИХ ТЕХНОЛОГІЙ .....	10
1.1 Завдання поширення інформації в мультиагентному середовищі .....	10
1.2 Управління навантаженням у розподілених системах та парадигма мобільних агентів .....	12
2 ДОСЛІДЖЕННЯ МЕХАНІЗМУ ПОШИРЕННЯ ІНФОРМАЦІЇ В МУЛЬТИАГЕНТНОЇ СИСТЕМИ .....	17
2.1 Побудова дерев агрегації даних для збільшення терміну служби мережі в багатоагентних системах .....	17
2.2 Архітектура складових мобільних додатків.....	110
3 МОБІЛЬНІ ПРОГРАМНІ АГЕНТИ ЯК ІНСТРУМЕНТ УПРАВЛІННЯ НАВАНТАЖЕННЯМ У РОЗПОДІЛЕНИХ ІНФОКОМУНІКАЦІЙНИХ СИСТЕМАХ .....	22
3.1 Мобільні агенти та балансування навантаження.....	22
3.2 Результати експерименту .....	28
3.3 Міграція послуг на основі мобільних агентів для забезпечення стійкості до зовнішніх та внутрішніх обурень.....	38
3.4 Прийняття рішень про міграцію локальних та спільних агентів .....	41
ВИСНОВКИ .....	48
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	50
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	53

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ  
І ТЕРМІНІВ

ІС – інформаційні системи

НА – Агент хоста

DA – Агент виявлення

RA – Агент повторного підключення

WSN – Бездротові сенсорні мережі

ІоТ – Інтернету речей

СУП – сервер управління подіями

СУР – сервер управління ресурсами

МА – мобільний агент

СВА – середовище виконання агента

П – платформа

ВВОЗ - виявлення вторгнень та оцінка збитків

МС – моніторинг системи

## ВСТУП

Актуальність теми. Стрімкий розвиток інфокомунікаційних служб (об'єктів розвинених складно-структурованих інформаційних систем) породило безліч методів управління ними. У теоретичному плані низка методів управління інфокомунікаційними службами зводиться до раціоналізації розподілу обчислювачів на вирішення зовнішніх завдань. Альтернативою тут є концепція мобільних агентів з розширенням алгоритмів балансування навантаження для поліпшення рівномірності її розподілу.

Незважаючи на розвинену теорію управління розподіленими службами інфокомунікаційних середовищ, існують проблеми, пов'язані з урахуванням працездатності виконавчих платформ та динамічною реконфігурацією служб параметрів систем. Отже, необхідно розробити алгоритм розширеного локального перегляду бездротової агентної мережі з фіксованим або випадковим приймачем даних на основі алгоритмів відновлення шляху зниження накладних витрат на взаємодію та вимог до повноти інформації про мережу на приймачі.

Таким чином, актуальність теми дослідження продиктована необхідністю подальшого розвитку засобів спеціального математичного та програмного забезпечення процесів керування інфокомунікаційними службами (об'єктами ІС) на основі реалізації систем мобільних агентів.

Метою роботи є розробка засобів спеціального математичного та програмного забезпечення процесів управління інфокомунікаційними службами (об'єктами ІС) на основі реалізації систем мобільних агентів.

Завдання дослідження. Для досягнення поставленої мети необхідно вирішити такі завдання:

- розробити алгоритми балансування навантаження для покращення рівномірності її розподілу, що ґрунтуються на використанні мобільних агентів;

- запропонувати архітектуру системи мобільних агентів з динамічним урахуванням коректності роботи платформ та динамічним переміщенням служб для підвищення доступності послуг;

- розробити алгоритм розширеного локального перегляду бездротової агентної мережі з фіксованим або випадковим приймачем даних на основі алгоритмів відновлення шляху для зниження накладних витрат на взаємодію та вимог до повноти інформації про мережу на приймачі;

- запропонувати архітектуру складових мобільних додатків із використанням високорівневого опису системи мобільних агентів та реалізацією процесу гетерогенної композиції мобільного додатка.

Об'єкт дослідження: процеси керування інфокомунікаційними службами (об'єктами ІВ).

Предмет дослідження: математичного та програмного забезпечення процесів управління інфокомунікаційними службами на основі систем мобільних агентів.

Отримано результати, що характеризуються науковою новизною. Архітектура складових мобільних додатків, відрізняється високорівневим описом системи мобільних агентів та забезпечує реалізацію процесу гетерогенної композиції мобільного додатка.

# 1 АНАЛІЗ УПРАВЛІННЯ РОЗПОДІЛЕНИМИ СИСТЕМАМИ НА ОСНОВІ МУЛЬТІАГЕНТНИХ ТЕХНОЛОГІЙ

## 1.1 Завдання поширення інформації в мультиагентному середовищі

У роботі досліджується процес поширення інформації в мультиагентній системі, в якому кожен агент представлений ланцюгом Маркова з двома станами - L та M. Стан L відноситься до «дому», тоді як стан M відноситься до «місця зустрічі». Коли два агенти залишаються разом в M, вони зустрічаються і утворюють контакт. Це означає, що можуть обмінюватися інформацією, здійснювати інші взаємодії. Припустимо, що спочатку всі знаходяться в стані L, і рівно один з агентів має інформацію. Загалом процес можна класифікувати як процес поширення з мобільними агентами, і він дозволяє продемонструвати кілька цікавих властивостей моделі. Надано ефективний спосіб обчислення часу поширення та досліджено залежність процесу поширення від таких параметрів, як кількість агентів, кількість неінформованих агентів в кінці процесу та інтенсивність контакту. Загальноприйнятий підхід до моделювання процесів поширення має на увазі мережу, в якій окремі елементи представлені вузлами, а сусідні вузли взаємодіють один з одним і обмінюються інформацією (використовуватимемо цю термінологію далі). У будь-який момент часу вузол, що містить інформацію, може відігравати роль розповсюджувача, розповсюджуючи інформацію серед своїх сусідів. Інші вузли можуть представляти необізнаних агентів (сприйнятливо-інфекційна модель) та поінформованих агентів, які не здатні до подальшої передачі інформації (сприйнятливо-інфекційна відновлена модель).

Існує безліч варіацій моделі, що включають випадковості, зміни статусу, як представлено в [2,3]. Центральні досліджувані питання включають час, витрачене на передачу інформації від вузла до інших вузлів

[5] або характеристику стану системи.

Зокрема, розглядаємо кінцеве безліч  $N$  мобільних вузлів (агентів), кожен з яких переміщається між своїми індивідуальними локаціями  $L_1, \dots, L_N$ . Загальне місце зустрічі позначається як  $M$ . Рух у мережі контролюється за допомогою безперервної мережі Маркова. Час, витрачений у стані  $L_i$ , експоненційно розподілено з параметром  $\lambda$ , після чого агент переходить в  $M$ . Тривалість часу  $M$  експоненційно розподілена з параметром  $\mu$ , після чого агент повертається у вихідний стан  $L_i$ . Спочатку всі  $N$  вузлів знаходяться біля своїх витоків; одному з них повідомляється інформація. Інформований агент на місці зустрічі  $M$  ділиться інформацією коїться з іншими агентами, що у  $M$ .

Цікавить спосіб розповсюдження інформації і, зокрема, час, доки всі вузли не будуть проінформовані, тобто час розповсюдження.

Базовий граф мережі є зіркоподібним графом з єдиним основним хаб-вузлом  $M$ , що представляє місце зустрічі, і листям  $L_1, \dots, L_N$ , що представляють локації агентів. Спростимо модель, об'єднавши джерела  $L_1, \dots, L_N$  в одну локацію  $L$ , в якій не створюються взаємодії та не передається інформація. Таким чином, замість розглядати зірковий граф, маємо просто мережу з двома вузлами  $M$  і  $L$ .

У моделях епідемій часто використовується безрозмірне репродуктивне число  $R = \lambda(k)/\mu$ , де  $\lambda(k)$  – швидкість, з якою людина з  $k$  контактами поширює хворобу, а  $\mu$  – індивідуальна швидкість одужання. Аналогічно використовуємо безрозмірну міру контактної інтенсивності  $\rho = \lambda/\mu$ , що дає відношення швидкості індивідуального переходу від  $L$  до  $M$  до швидкості зворотного руху.

Велика інтенсивність контактів означає, що час перебування в  $M$  значно більше часу, проведеного в  $L$ , а інтенсивність взаємних контактів висока. Загальноприйняті моделі процесів поширення суттєво відрізняються від аналізованих. Ці процеси або остигають стійкого стану, або закінчуються в одному поглинаючому стані (наприклад, коли всі люди виліковуються). У

нашій моделі є один поглинаючий стан, коли всі вузли поінформовані. Модель, що представляється, належить до сімейства процесів поширення з рухливістю.

## 1.2 Управління навантаженням у розподілених системах та парадигма мобільних агентів

Розподіленим системам необхідно виконувати балансування навантаження на своїх хостах, щоб обчислення виконувались якнайшвидше. Дослідження в цій галузі показали появу парадигми мобільних агентів як перспективне рішення. У цій роботі ця парадигма використовується, щоб запропонувати підхід до балансування навантаження, який використовує переваги мобільності агентів, зокрема, на етапі збору інформації. Необхідно мати загальне системне бачення при одночасному зниженні витрат на мережевий зв'язок, а також інших переваг, таких як стійкість до відмов і розширюваність для великомасштабних мереж. Таким чином, мета полягає в тому, щоб покращити розподіл навантажень збалансованим чином, щоб максимально наблизити навантаження до середнього навантаження системи. Результати експериментів показують ефективність пропонованого підходу до балансування навантажень та скорочення часу відгуку.

Тенденція комп'ютерного світу до " розподіленим системам " більше передбачає роботу одного комп'ютера без взаємодії чи співробітництва з іншими комп'ютерами. Ці системи мають бути спроектовані з урахуванням нових вимог. Технологія мобільного програмного агента є однією з відомих технологій у галузі розподілених обчислень. Він виник як альтернатива класичній парадигмі "клієнт/сервер", яка є найбільш широко використовуваним підходом при створенні розподілених додатків [5]. Технологія мобільних агентів має цікаві перспективи для різних сфер застосування, серед яких ми знаходимо електронну комерцію, пошук інформації в Інтернеті та область балансування навантаження (LB).

Досягнення кращої продуктивності у розподілених системах проблема балансування навантаження інтенсивно вивчалася дослідниками [7, 8]. Балансування дозволяє максимально використовувати доступні ресурси і це може бути досягнуто за рахунок розумного розподілу завдань.

При динамічному балансуванні навантаження P2P-мережі малого світу з використанням мобільних агентів пропонується, по-перше, об'єднувати однорангові вузли, які мають однаковий набір загальних ресурсів. Потім вони балансують навантаження запитів на вузлах усередині кластера, щоб уникнути перевантаження мережі. Для цієї мети визначено три агенти: Агент хоста (HA), Агент виявлення (DA) та Агент повторного підключення (RA). DA мігрує між внутрішніми вузлами для виявлення скупчень вузлів. Цей агент генерується та контролюється HA, який проходить через усі групи. RA генерується для повторного підключення зв'язків між перевантаженими вузлами та недовантаженими вузлами одного й того кластера. Параметр привабливості вимірюється для кожного вузла відповідно до деяких критеріїв, таких як ступінь вузла, продуктивність обробки та ресурси, що містяться у вузлі. DA мігрує на вузол з високим параметром привабливості, коли DA прибуває, він перевіряє, чи перевантажений вузол або недовантажений. Однак алгоритм вибору вибирає партнерський вузол, який може відмовитись приймати запити агента DA, якщо він перевантажений. У цьому випадку агент DA мігрує на інші вузли, доки не знайде потрібний вузол. Таким чином, правильний партнер може бути неправильно вибраний до початку процесу міграції.

В алгоритмі балансування навантаження для гетерогенних P2P-систем використовується один тип мобільного агента з його основними компонентами, такими як збирання, аналіз та місцезнаходження. Коефіцієнт використання вимірюється відповідно до навантаження та пропускної здатності кожного вузла. Однак для перенесення навантаження автори пропонують вибрати сусідній вузол з найменшим коефіцієнтом використання. Таким чином, цей один недовантажений вузол буде обраний

більшістю перевантажених вузлів чи всіма з них.

Отже, вибраний вузол буде перевантажений або відмовиться від прийому завдань, що призведе до повторного запуску процесу позиціонування та повторної спроби з іншими вузлами, таким чином, більше часу витрачається на позиціонування, міграцію та виконання.

Враховуючи зростаючу складність та динамічність сервіс-орієнтованих систем, забезпечення доступності послуг, незважаючи на шкідливі атаки або системні збої, є складним завданням. У додатках, де доступність послуг має вирішальне значення, стратегічна міграція служб, що залежать від часу, з їхньої поточної платформи на інші платформи є життєздатним рішенням, що дозволяє забезпечити безперервне надання цих послуг на нових платформах. Міграція служб потребує спеціальної інфраструктури та механізмів для переміщення сервісних програм та даних з однієї платформи на іншу. Розробка ефективної схеми міграції з необхідною системною підтримкою є важливою для забезпечення гарантованої міграції послуг.

Були проведені інтенсивні дослідження з міграції процесів, завдань та послуг для кількох цілей, включаючи високу продуктивність [3], відмовостійкість [15], адаптивність системи та гнучкість ресурсів [7], безпеку та живучість [9]. Однак міграція послуг на основі мобільних агентів є перспективною галуззю досліджень. Агенти – це автономні та соціальні програмні об'єкти, використовувані розробки складних додатків, надають послуги користувачам [10]. Ідея, що лежить в основі концепції “агента”, полягає в активному та автономному модулі, який може співпрацювати та/або конкурувати з іншими модулями/агентами та навколишнім середовищем [13].

У разі шкідливої атаки та збою системи мобільні агенти можуть перейти зі своїх пошкоджених платформ на інші справні платформи, щоб запропоновані ними послуги могли постійно надаватися на нових платформах, забезпечуючи цим міграцію послуг. Міграція служб за допомогою стратегічного переміщення агентів допомагає системі пережити

пошкодження хоста та підвищує доступність служб. У роботі описується заснований на мобільних агентах підхід до міграції служб, у якому група агентів спільно формує план міграції для переміщення зі своїх поточних платформ інші більш безпечні і надійні платформи.

Визначимо системну архітектуру підтримки міграції агентів і опишемо модель спільного прийняття рішень групи агентів у процесі міграції. У роботі представлено схему міграції послуг на основі мобільних агентів. У випадку, якщо деякі платформи в системі були пошкоджені внаслідок інциденту безпеки або деякі компоненти цих платформ функціонально вийшли з ладу, мобільні агенти можуть динамічно переміщатися зі своїх поточних платформ на інші справні, більш надійні платформи і продовжувати свою роботу на цих нових платформах, таким чином переживши інцидент безпеки чи збій системи. План спільної міграції гарантує, що міграція агента не порушить жодних операційних обмежень групи агентів. Оскільки агенти є соціальними суб'єктами, деякі агенти можуть спільно працювати з іншими над одним і тим самим завданням.

Один агент може залежати функціонально від інших агентів. Запропонований підхід ефективно перевіряє, що план спільної міграції не порушуватиме жодне з обмежень. Щоб продемонструвати здійсненність схеми міграції агентів, розроблена система мобільних агентів, що підтверджує концепцію, засновану на існуючій платформі мобільних агентів.

Хоча моделювання проводилося на основі невеликої групи агентів, воно показує потенціал застосування запропонованої схеми до більш складних програм для підвищення доступності послуг.

Агентні технології все частіше використовуються в різних програмах, і мобільні агенти, зокрема, можуть підтримувати адаптивність системи та доступність послуг. У цьому розділі коротко викладено пов'язані з цим роботи з міграції мобільних агентів.

Метод скорочення трафіку даних у результаті міграції агентів передбачає зосереджену роботу на середовищі виконання агента, яка кешує

код агента та статус агента, щоб код і статус можна було використовувати повторно при поверненні мобільного агента. Запропонований підхід дозволяє гнучко впроваджувати мобільні агенти.

## 2 ДОСЛІДЖЕННЯ МЕХАНІЗМУ ПОШИРЕННЯ ІНФОРМАЦІЇ В МУЛЬТИАГЕНТНОЇ СИСТЕМИ

2.1 Побудова дерев агрегації даних для збільшення терміну служби мережі в багатоагентних системах

Збільшення часу життя мережі (NL) є важливою вимогою у бездротових сенсорних мережах (WSN). Одним із методів збільшення NL є використання дерев агрегації даних (DAT). DAT покращують NL, комбінуючи переваги енергоефективності як Агрегації даних (DA), так і маршрутизації на основі дерева. Поки що для побудови DAT широко використовуються як централізовані, так і розподілені стратегії, ми пропонуємо комбінований підхід для покращення NL. Цей підхід знижує накладні витрати на зв'язок та послаблює потребу у повній інформації про мережу на приймачі. У цій роботі такий комбінований підхід називається підходом розширеного локального перегляду (ELV). На основі ELV запропоновано два алгоритми побудови DAT: ELV з фіксованим приймачем (ELVF) та ELV з випадковим приймачем (ELVR). Обидва алгоритми використовують заснований на евристиці метод відновлення локального шляхи (LPR) та жадібний алгоритм розширеного відновлення шляху (EPR).

Використовуючи ці методи планується послідовність DAT, які в сукупності покращують NL, а також зменшують накладні витрати на реструктуризацію DAT. Ефективність ELVF та ELVR оцінюється з допомогою строгих експериментів і результати моделювання показують, що запропоновані алгоритми покращують NL і масштабуються при різних значеннях коефіцієнта DA.

Бездротові сенсорні мережі (WSN) – це інформаційні та комунікаційні технології для Інтернету речей (IoT), що з'єднують все в розумному світі. У WSN вузли-датчики збирають інформацію, встановлюють шлях до приймача

і надсилають дані цим шляхом. Залежно від вимог програми та можливостей вузлів-датчиків, вузли безпосередньо надсилають дані до приймача або обмінюються даними через інші вузли, доки дані не досягнуть приймача.

Оскільки зв'язок між вузлами споживає більше енергії, ніж обчислення та вимірювання [1], протоколи маршрутизації, які зменшують кількість передач через мережу, підходять для використання в умовах обмеженого енергоспоживання. Скорочення числа передач здійснюється застосуванням агрегації даних (DA) – відомого методу підвищення енергоефективності [2, 4]. Маршрутизація на основі дерева агрегації даних (DAT) у WSN поєднує переваги як DA, так і маршрутизації на основі дерева та забезпечує енергоефективну стратегію маршрутизації, яка продовжує час життя мережі (NL).

Час життя мережі NL – це час, протягом якого всі вузли мережі можуть передавати дані до приймача. DAT підходять для сценаріїв зі строгими енергетичними умовами та агресивними та критичними середовищами, де втручання людини неможливе. Економічна ефективність DAT визначається структурою DAT та коефіцієнтом DA ( $\alpha$ ). Структура DAT визначає шляхи зв'язку вузлів передачі даних у приймач, а коефіцієнт DA визначає, скільки блоків даних об'єднується у вузлі [4,6]. DAT. На рисунку 2.1 показано три підходи до побудови DAT. У першому підході (рисунок 2.1а) приймач має повну інформацію про мережу і шляхи маршрутизації розробляються самим приймачем з використанням централізованих глобальних методів. У другому підході (рисунок 2.1б) кожен вузол взаємодіє з вузлами у своєму діапазоні передачі та визначає свої шляхи, використовуючи локальні методи. У третьому підході (рисунок 2.1в) приймач має часткову інформацію про мережу. Шляхи маршрутизації складаються з використанням комбінації централізованих глобальних методів у приймачі та локальних методів в окремих вузлах. Цей підхід можна застосовувати на практиці, оскільки він знижує накладні витрати на зв'язок та зменшує потребу у повній мережній інформації на приймачі.

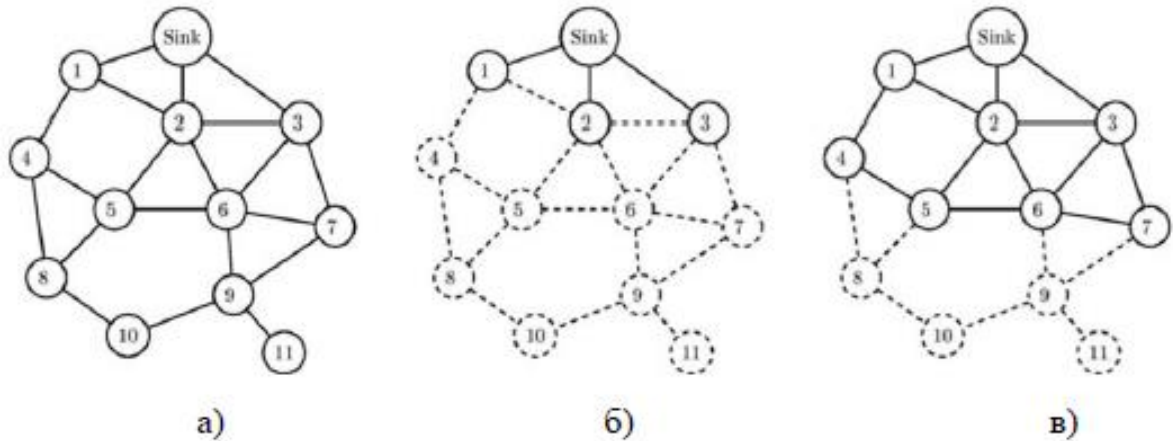


Рисунок 2.1 – Підходи до побудови DAT

У роботі спільний підхід називається підходом розширеного локального перегляду (ELV), і він доводить запропоновані методи розробки стратегій побудови DAT з його основи.

Максимізації NL сприяє ефективне управління енергоспоживанням з використанням підходів маршрутизації на основі дерева агрегації даних [4.2]. У цьому випадку NL збільшується за рахунок створення DAT, у яких всі вузли рівномірно та оптимально споживають енергію. Це NP-повне завдання [4, 7] і кілька дослідників розробили методи побудови DAT, які неоптимально розподіляють енергоспоживання вузлів [6, 7].

## 2.2 Архітектура складових мобільних додатків

Використання та розробка мобільних додатків зростають у геометричній прогресії. Мобільна ніша представляє нові проблеми розробки програмного забезпечення [4]. Мобільні платформи швидко змінюються, включаючи різноманітні можливості, такі як GPS, датчики та режими введення. Для цього мобільні додатки повинні мати можливість спеціалізувати свою поведінку відповідно до контекстної інформації і, отже, мають бути багатоканальними та працювати на всіх платформах. Більше

того, незважаючи на велику кількість доступних мобільних додатків, потреби користувачів у повсякденному житті відрізняються одна від одної. Ці проблеми створюють деяку необхідність у механізмі композиції мобільних додатків для досягнення бажаних функціональних можливостей з урахуванням різної контекстної інформації (тобто характеристик програмного та апаратного забезпечення мобільного пристрою). Використовуючи механізм композиції, можна вирішити ключове завдання розробки програмного забезпечення – можливість повторного використання об'єктів [5].

Наскільки відомо, не існує такого механізму композиції мобільних додатків, який би враховував різні фактори, такі як вид об'єктів програмного забезпечення, характеристики мобільних пристроїв, поточний стан мобільних пристроїв і вартість композиції. Таким чином, використовуючи переваги існуючих робіт, у роботі подано композиційний процес для створення контекстно-залежних гетерогенних мобільних додатків з урахуванням вартості композиції. Основна мета: по-перше, створювати мобільні програми з використанням існуючих гетерогенних програмних об'єктів шляхом забезпечення процесу гетерогенної композиції, щоб максимізувати переваги існуючих функціональних можливостей та, таким чином, задовольнити вимоги користувачів.

По-друге, облік контекстної інформації мобільного пристрою під час композиції, щоб розробляти мобільні програми, які можуть сприймати та адаптуватися до середовища їх виконання. Робота є елементом процесу створення контекстно-залежних додатків на основі гетерогенних об'єктів за допомогою підходу метамодельовання в мобільному середовищі.

Ці проблеми створюють деяку необхідність у механізмі композиції мобільних додатків для досягнення бажаних функціональних можливостей з урахуванням різної контекстної інформації (тобто характеристик програмного та апаратного забезпечення мобільного пристрою). Використовуючи механізм композиції можна вирішити ключове завдання

розробки програмного забезпечення: можливість повторного використання існуючих об'єктів [5].

Наскільки відомо, не існує такого механізму композиції мобільних додатків, який би враховував різні фактори, такі як вид об'єктів програмного забезпечення, характеристики мобільних пристроїв, поточний стан мобільних пристроїв і вартість композиції. Таким чином, використовуючи переваги існуючих робіт, у роботі подано композиційний процес для створення контекстно-залежних гетерогенних мобільних додатків з урахуванням вартості композиції. Основна мета: по-перше, створювати мобільні програми з використанням існуючих гетерогенних програмних об'єктів шляхом забезпечення процесу гетерогенної композиції, щоб максимізувати переваги існуючих функціональних можливостей та, таким чином, задовольнити вимоги користувачів.

## 3 МОБІЛЬНІ ПРОГРАМНІ АГЕНТИ ЯК ІНСТРУМЕНТ УПРАВЛІННЯ НАВАНТАЖЕННЯМ У РОЗПОДІЛЕНИХ ІНФОКОМУНІКАЦІЙНИХ СИСТЕМАХ

### 3.1 Мобільні агенти та балансування навантаження

Технологія мобільних агентів забезпечує підтримку балансування навантаження з трьома основними характеристиками: вони можуть переміщатися з однієї платформи до іншої, вони можуть переміщатися між різнорідними платформами, характеру (наприклад, операційна система, ємність процесора, сховище тощо), вони містять код конкретної програми, замість того щоб вимагати попереднього встановлення цього коду на цільовій машині.

Дві основні переваги, які дає підхід мобільного агента для балансування навантаження, полягають у наступному: скорочення мережного трафіку, міграція процесів з одного сайту на інший.

У цьому розділі запропоновано внесок у балансування навантаження на основі парадигми мобільного агента. Надалі припускаємо, що вузли перебувають у «кооперативній» мережі, що дозволяє нам вивчати лише характеристики аналізованого алгоритму, і передбачається, що незалежні, тобто. між ними немає зв'язку.

Для визначення стану кожної машини необхідно визначити порогові значення: існують статичні порогові значення, які є незмінними фіксованими значеннями та інші динамічні граничні значення, які змінюються залежно від еволюції системи. Використовується другий варіант, у якому граничне значення визначається відповідно до середнього навантаження системи, що більше підходить для динамічного середовища.



Рисунок 3.1 – Роль кожного запропонованого агента

Запропоновано три типи агентів: стаціонарного агента та двох мобільних агентів (рисунок 3.1): агент-спостерігач (ОА), мобільний агент-супервайзер (SMA), мобільний агент-транспорттер (ТМА). Далі наведено виклад переваг кожного запропонованого процесу у запропонованому підході, показуючи різні випадки, що можуть виникнути. Дослідження переваг запропонованого процесу.

ОА<sub>i</sub> бачить недовантаженість власного сайту. Надсилається хвильове повідомлення про стан, і перевантажений ОА<sub>j</sub> інформує ОА<sub>i</sub> про це. Тоді можливі варіанти:

- 1 ОА<sub>i</sub> довантажується від ОА<sub>k</sub> (запит не приймається);
- 2 додаткове навантаження ОА<sub>j</sub> перевантажить ОА<sub>i</sub> (запит не приймається);
- 3 ОА<sub>i</sub> довантажується від ОА<sub>j</sub> (запит не приймається).

ТМА переміщається між вузлами від  $S_j$  у пошуках вузла для перенаправлення завантаження та знаходить потрібний вузол  $S_z$ . ОА<sub>j</sub> просто вибирає ОА<sub>z</sub> із глобально мінімальним завантаженням.

Розглянемо детально кожного з запропонованих агентів.

Агент-спостерігач. Мета нашого алгоритму полягає в тому, щоб привести систему до збалансованій стан і, отже, мати збалансовані машини. Роль ОА полягає у спостереженні за навантаженням машини, на якій вона розташована.

Вимірювання локального навантаження вузла. Існує два методи вимірювання навантаження: один - на вимогу, а інший – періодичний. На вимогу у цьому випадку ОА вимірює навантаження на вузол, коли SMA просить його зробити це (тобто коли SMA надходить на цей вузол. Таким чином, ОА повинен знати час прибуття SMA, що означає, що SMA повинен повідомити ОА сигнальним повідомленням. В іншому випадку SMA повинен почекати, поки ОА не обчислить локальне навантаження, що збільшує час збору навантаження SMA.

При періодичному у цьому випадку ОА не потрібно знати час прибуття

SMA, так як він вимірює локальне завантаження свого сайту кожен період  $T_{info}$ . Коли SMS надходить на цей сайт, OA надає йому останнє вимірне значення навантаження. Інформація має бути вказана за відповідний період, щоб уникнути перевантаження системи. Якщо навантаження швидко змінюються, то інформація має бути невеликою. У в іншому випадку необхідно збільшити тривалість  $T_{info}$ .

Будемо використовувати другий метод, тому що він простий у реалізації, не вимагає віддаленого обміну повідомленнями між SMA та OA та скорочує час, що виділяється на політику збору інформації.

Далі йде запуск процесу міграції. Коли OA сайт  $S_j$  отримує повідомлення від SMA, він отримує повідомлення про те, що його сайт перевантажений. Тому цей OA запускає агента TMS, вказуючи надлишок для транспортування, використовуючи середню навантаження, відправлене SMS. Крім того, кінцевий сайт оголошено в повідомленні SMA. Така передача навантаження між двома незбалансованими машинами наближає їх навантаження до середнього навантаження системи, щоб зробити їх більш збалансованими (рисунок 3.2).

```

For each site  $S_k$  {
  For each period  $T_{info}$  {
    Measure local load  $L_k$  //  $L_k$  is the Local load of site  $S_k$ 
  }
  Receive_info(SMA,  $S_r$ , AL); // SMA indicates the recipient site  $S_r$ .
  // AL is the average load
  If (Received_info != NULL) {
    Launch the TMA agent and send it to the site  $S_r$ 
  }
  Else //  $S_k$  is not overloaded
  OA declares: "My site is lightly loaded"
  OA declares: "I am waiting for the reception of transporter agents from
other sites"
}
}

```

Рисунок 3.2 – Сценарій агента-спостерігача

Мобільний агент-супервайзер. SMA циклічно переміщається між різними вузлами мережі. Для кожного сайту SMA витягує ім'я сайту та його значення завантаження та поміщає в таблицю (tab\_system). Закінчивши цей рух, SMA робить два наступні кроки. Розрахунок середнього навантаження системи (AL). Після прибуття агента SMA на сайт він зв'язується з ОА, щоб отримати локальне навантаження. Потім SMA додає це значення навантаження до вже зібраних значень. Коли SMA завершує свою траєкторію, він обчислює середнє навантаження "AL". Ця AL забезпечує:

- досягнення глобального балансування навантаження, оскільки навантаження на кожен вузол порівнюється із загальним навантаженням системи;
- управління змінами навантаження, які можуть виникнути в системі, чого відбувається щодо фіксованого порога.

При визначенні стану вузла та побудови двох таблиць використовуємо середнє навантаження (AL) як "порог", в відповідно до якого ми можемо розрізнити 3 стани:

- сайт перевантажений, якщо його завантаження  $> AL$ ;
- сайт не перевантажений, якщо його завантаження  $< AL$ ;
- нейтральний стан, якщо навантаження  $= AL$ .

З таблиці Tab\_System та розрахованого середнього навантаження агент SMA будує дві підтаблиці без вкладок (містить недовантажені сайти) та Tab\_plus (містить перевантажені сайти).

Після цього SMA упорядкує значення навантаження для кожної з цих таблиць. У таблиці без табуляції воно проходить у порядку зростання (від найменшого значення навантаження до найбільшого). А у таблиці Tab\_Plus: у порядку зменшення.

Далі йде вибір партнера. Після цих двох кроків створюється глобальне бачення спільної картини, і тепер SMA відіграє роль планувальника. Він визначає для кожного перевантаженого сайту партнера (недовантажений сайт одержувача) таким чином, щоб перевантажений сайт та його партнер мали

однаковий індекс таблиці відповідно до `tab_plus` та `tab_less`. У докладному вигляді запропонований для вибору партнера метод полягає в тому, щоб призначити перший запис `Tab_Less` першого запису `Tab_Plus`. Це робиться для того, щоб призначити сайт із найбільшою навантаженням сайту з найменшим навантаженням та призначити другий запис `Tab_Less` другий запис `Tab_Plus` тощо. Кожний недовантажений сайт `Tab_Less` є партнером перевантаженого сайту `Tab_Plus` (рисунок 3.3).

```

Som=0 ; AL=0 ;
For m = (from 1 to nbSites) { // nbSites is the number of machines or
nodes
Request the local load from the OA of site Sj
Add(site name Sj + Lj load) as an entry in tab_system
Som=Som+Lj // Lj: local load provided by agent OA of site Sj
doMove (Sj, Sk) // SMA moves to the next node
}
AL = Som / nbSites ;
While Tab_Sys[i] not_empty {
If (Tab_System [i] > AL)
Add this site as an entry to tab_plus
Else {
If (Tab_Systeme [i] < AL)
Add this site as an entry to tab_less
}
i++; //next hut in tab_sys
}
Ordering Tab_Less in ascending order;
Ordering Tab_Plus in descending order;
While tab_plus [i] not_empty { //indicating partners
So = Tab_Plus[i] // source overloaded site
Su = Tab_Moins[i] //receiver underloaded site
Send_info (Sk, Sj ,AL) // SMA informs the overload site by the receiver
of its surplus load
}
}

```

Рисунок 3.3 – Скрипт мобільного агента-супервізора

Мобільний агент має бути швидким під час свого переміщення, щоб інформація про завантаження системи не змінювалася в час цього переміщення, тобто. агент повинен бути оновлений, таким чином, розмір коду SMA повинен бути зведений до мінімуму.

У запропонованій архітектурі можна використовувати один агент-

супервізор, але завжди можемо розглянути можливість збільшення числа цих агентів, і це залежить від розміру мережі (кількості учасників машин). У цьому випадку необхідно забезпечити механізм співробітництва між різними агентами-супервізорами, щоб сформувати глобальне бачення навантаження систему.

Мобільний агент-транспортер (рисунок 3.4). Це мобільний агент, запущений агентом перевантаженого ОА сайту, і він є надлишок цього останнього.

```
doMove(Sk,Sj) ;
// TMA moves from the overloaded site to the receiver site
Send (L_Plusk, Sk, Sj) ; // Sk : the source site name
// L_Plusk : excess load of site Sk
// Sj: the recipient site name (indicated in SMA script)
```

Рисунок 3.4 – Сценарій мобільного агента-транспортера

Метою ТМА є транспортування цього надлишку на сайт одержувача, на якому буде виконуватись завдання (завдання). Це агент ОА, який вказує ТМА на роботу, яку потрібно виконати. Ця робота є набором завдань. Кількість агентів ТМА залежить від того, скільки завдань вони можуть виконати. Огляд запропонованої системи зі сценарієм трьох агентів на 4 сайтах представлено на рисунку 3.5.

### 3.2 Результати експерименту

Стан навантажень. Для тестування поведінки та оцінки ефективності запропонованого алгоритми підходу агентів розробляються з використанням Java Eclipse IDE поверх платформи агента JADE. Використовується "FSMBehaviour" для агентів ОА та SMA. Така поведінка служить для уявлення складних завдань, воно відповідає потребам складного поведінки цих двох агентів.

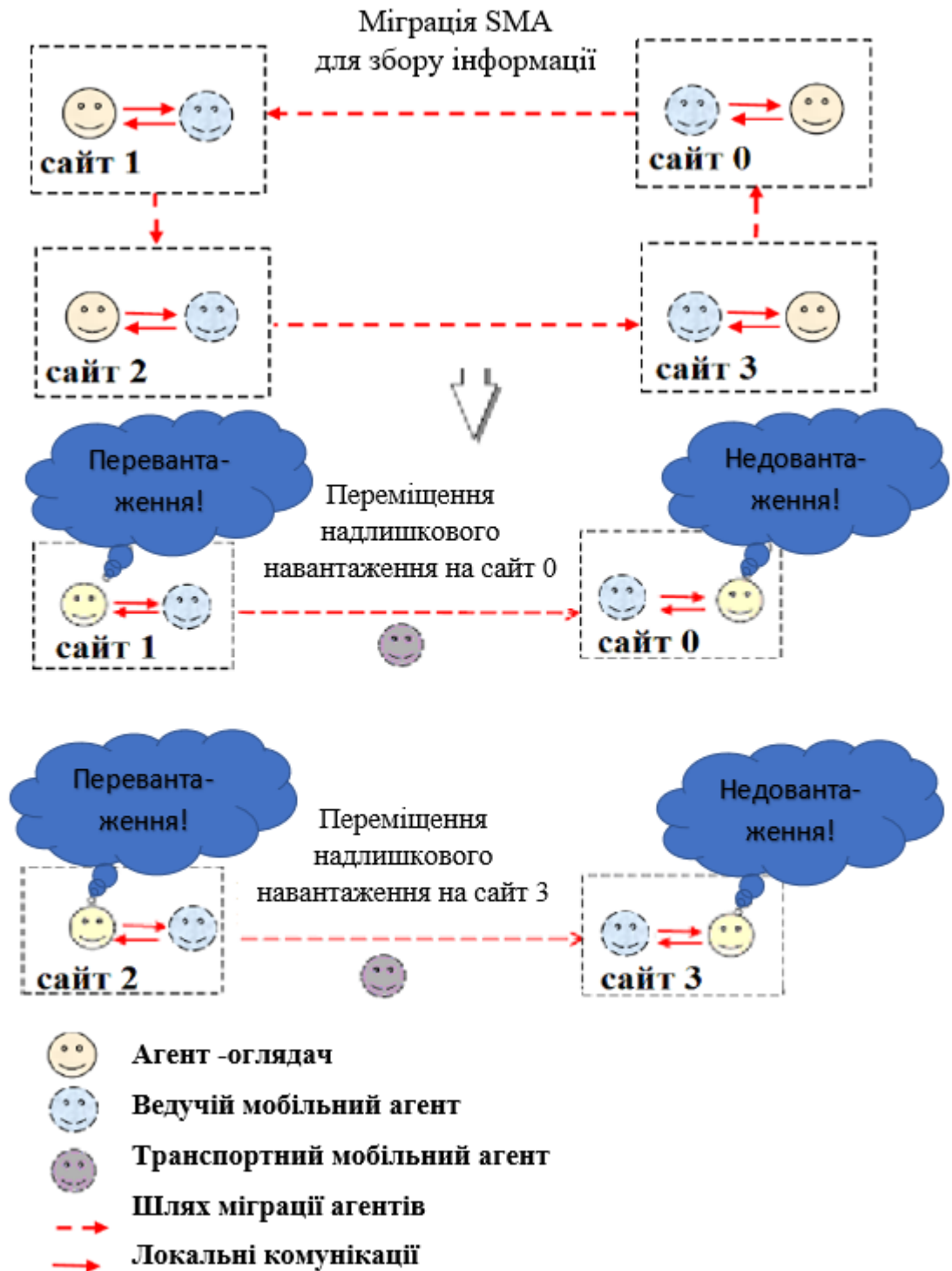


Рисунок 3.5 – Архітектура запропонованої системи

Для третього агента ТМА, який має тільки одне завдання для виконання, проста поведінка задовольняє потреби, тому в даному випадку це "OneShotBehaviour". Початковий код складається з трьох класів та їх підкласів відповідно до ролей на рисунку 3.1. Щоб запустити виконання розробленого алгоритму Jade, спочатку запускаються агенти-спостерігачі

(реалізовано вісім агентів  $OA_0, OA_1, \dots$ ), кожен у контейнері. Потім запускається агент SMA головний контейнер. Потім агент АМТ запускається відповідним агентом ОА (агентом перевантаженого сайту).

Коефіцієнт завантаження надається випадковим чином кожному контейнеру. На рисунках 3.6–3.9 представлено стан навантажень у три етапи: до застосування LB та при застосуванні пропонованого LB з табличним підходом та без нього.

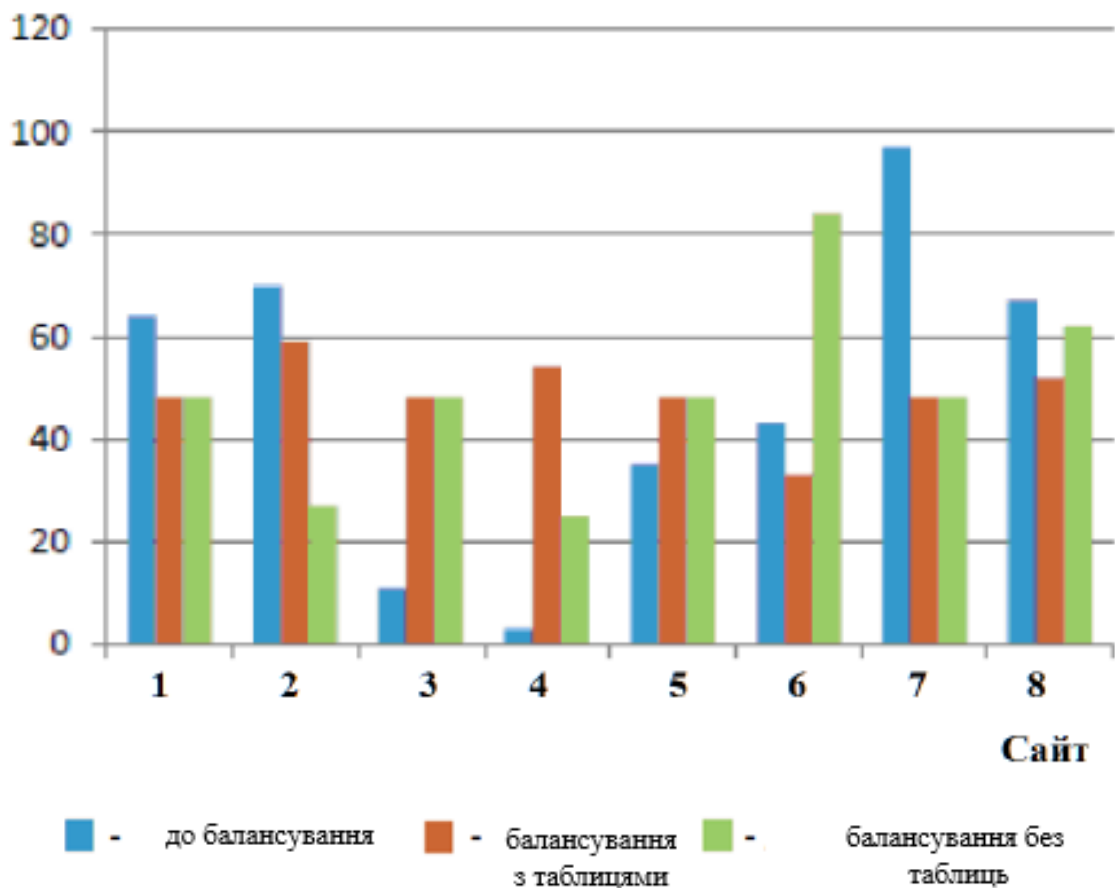


Рисунок 3.6 – Стан системи: вихідний

На третьому етапі (без таблиці) ТМА транспортує перевантаження на першу знайдену недовантажену ділянку.

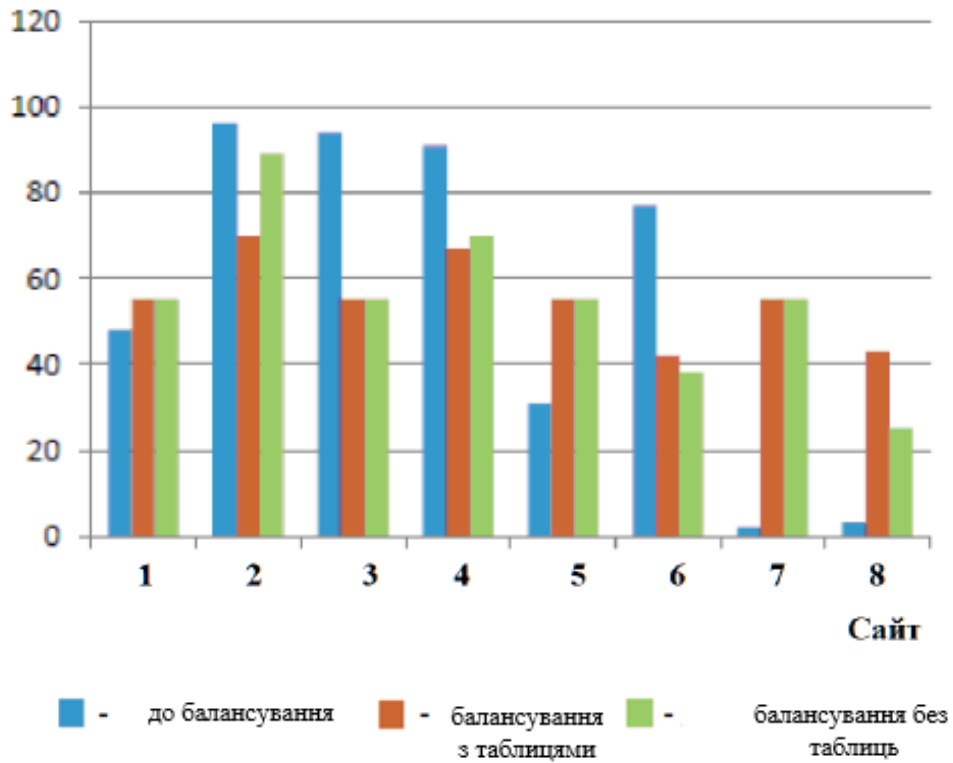


Рисунок 3.7 – Стан системи: перший етап

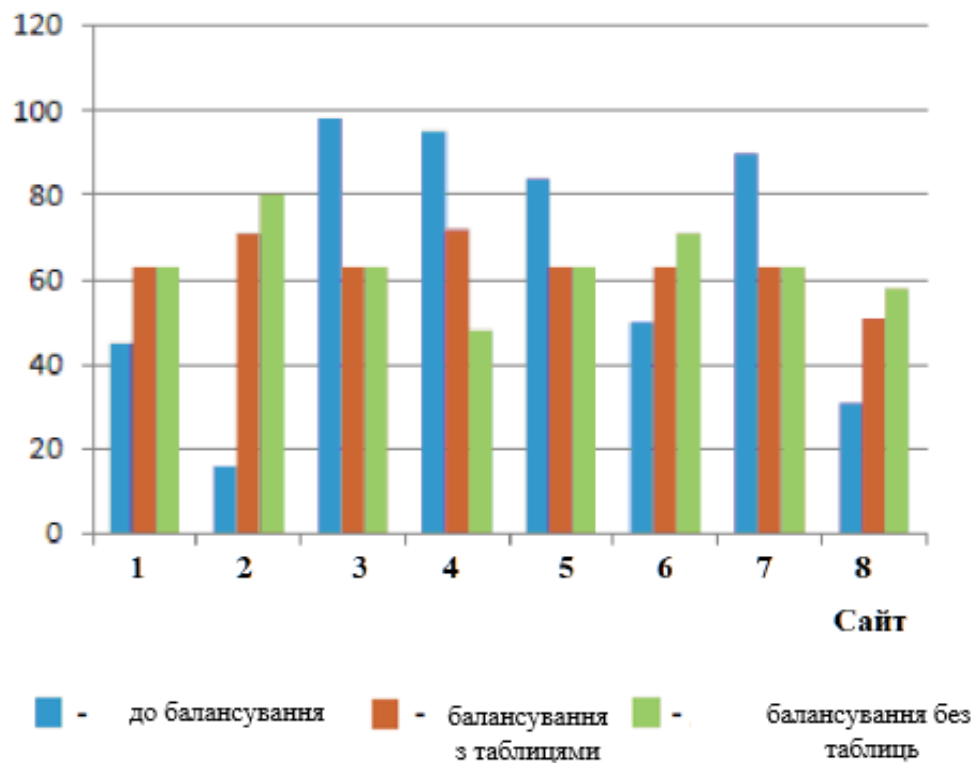


Рисунок 3.8 – Стан системи: другий етап

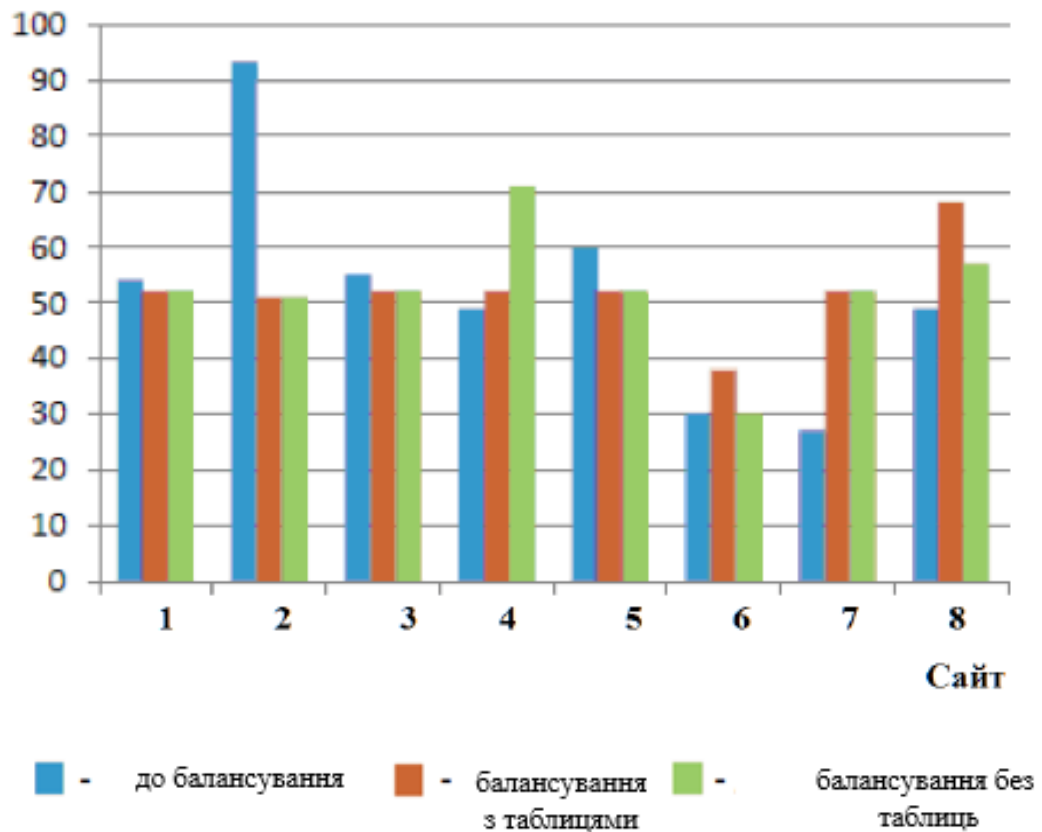


Рисунок 3.9 – системи: третій етап

На рисунку 3.6 показано, що першому етапі навантаження незбалансовані (від дуже високих до дуже низьких навантажень). На другому етапі (рисунок 3.7) навантаження починають балансуватися, що вказує вплив підходу LB. Після цього на третьому етапі навантаження (рисунок 3.8) більш збалансовані порівняно із загальним вихідним станом системи, що демонструє ефективність пропонованого табличного підходу при виборі правильного партнера для отримання надлишкового навантаження.

У розподілених обчисленнях час відгуку є суттєвою проблемою, і факт його скорочення є дуже важливим вимогою у підходах до вдосконалення. Однак за великих навантаження час відгуку системи збільшується. Серед такого типу набір завдань розподіляється між деякими вузлами системи. Час відгуку системи - це час, необхідний всім вузлам, що беруть участь реалізації цього набору завдань.

Час відгуку вимірюємо, використовуючи таку формулу (3.1):

$$T = (\text{NbTasks} \times \text{HighL} \times \text{ReqAT}) / 100 \quad (3.1)$$

де - NbTask - загальна кількість завдань,

- HighL – найбільший відсоток завантаження в цій системі,

- ReqAT – необхідний середній час для виконання одного завдання.

На рисунках 3.10 –3.12 показано зміну часу відгуку на трьох етапах при одночасному збільшенні числа завдань, які має виконувати ця система (між 200, 300, 400 та 500) для трьох наборів випадкових робітників навантажень. Результати доводять ефективність запропонованого алгоритму LB у скороченні часу відгуку порівняно з двома іншими фазами.

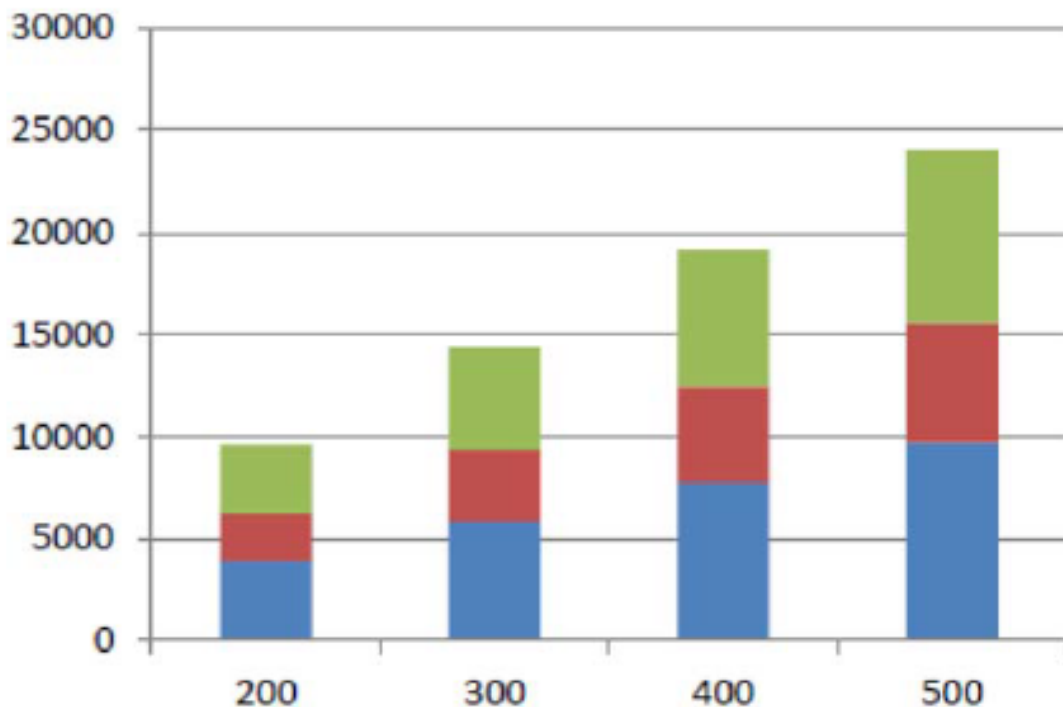


Рисунок 3.10 – Час відгуку як функція від кількості завдань  
(набір випадкових робочих навантажень №1)

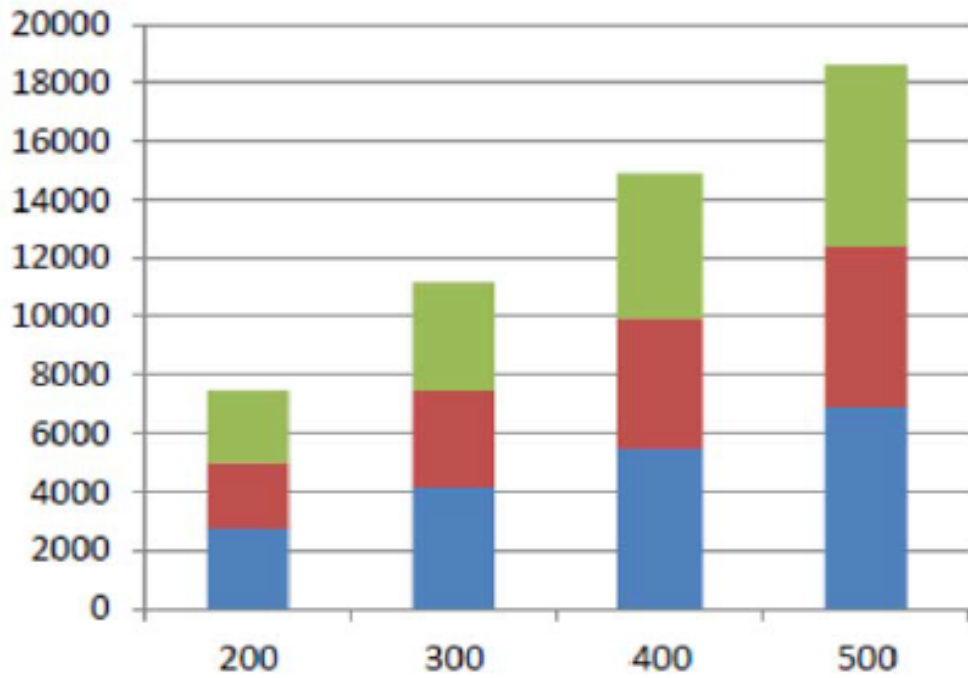


Рисунок 3.11 – Час відгуку як функція від кількості завдань (набір випадкових робочих навантажень №2)

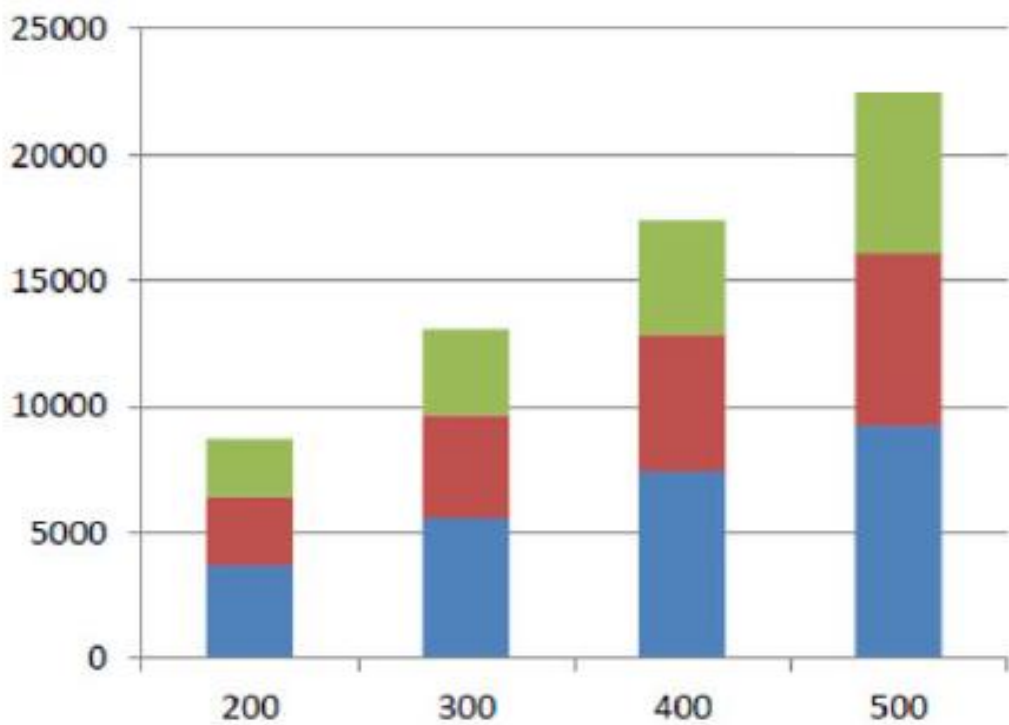


Рисунок 3.12 – Час відгуку як функція від кількості завдань (набір випадкових робочих навантажень №3)

Основна мета застосування процесу LB у розподіленій системі полягає

в тому, щоб усі вузли працювали з майже рівними робітниками навантаженнями. Іншими словами, зводиться до мінімуму різниця у частках робочого навантаження між вузлами і, таким чином, виключається наявність вузла з високим робочим навантаженням, у той час як інший вузол недостатньо завантажений. У розподіленій системі, де вузли працюють спільно для виконання певного набору завдань, балансування робочого навантаження і, отже, завантаження між вузлами має важливе значення для скорочення часу виконання (або часу відгуку) завдання та, отже, всіх запущених завдань. Тому в системі вимірюється різниця в показниках робочого навантаження між найбільш завантаженими вузлами та найменш завантаженими.

У запропонованій методиці вибору перевантажений вузол вибирає один із недовантажених вузлів, куди передавати перевантаження (партнера).

Методика вибору [14] ґрунтується на сортуванні недовантажених вузлів у списку відповідно до параметра привабливості. Далі список сортується відповідно до коефіцієнта використання. Потім кожен перевантажений вузол використовує цей список, щоб знайти найменш заряджений вузол передачі йому навантаження. Коли агент міграції прибуває на обраний вузол, він перевіряє, чи цей перевантажений вузол, щоб прийняти отримане навантаження або відхилити її, якщо вона недостатньо завантажена. У разі відхилення агент міграції переміщається на наступний вузол із відсортованого списку.

На рисунку 3.13 представлена різниця у показниках робочого навантаження між найбільш завантаженим сайтом (вузлом) та найменш завантаженим сайтом у кожному моделюванні (випадку). Цей параметр порівнюється в три етапи: перший етап - перед застосуванням LB, другий - при застосуванні пропонованого нами підходу та третій - при застосуванні LB з використанням методів вибору [14]. Рівень робочого навантаження присвоюється кожному вузлу випадковим чином, і запущено вісім симуляцій (8 випадків). Результати на рисунку 3.13 показують, що відмінності в

показники робочого навантаження більшою мірою зменшуються після застосування відомого підходу, а потім значно зменшуються при застосуванні запропонованого підходу. Ці результати доводять ефективність запропонованої методики вибору порівняно з існуючими методами вибору найбільш відповідного недовантаженого вузла для кожного перевантаженого вузла, і, таким чином, всі вузли працюють із майже однаковим навантаженням.

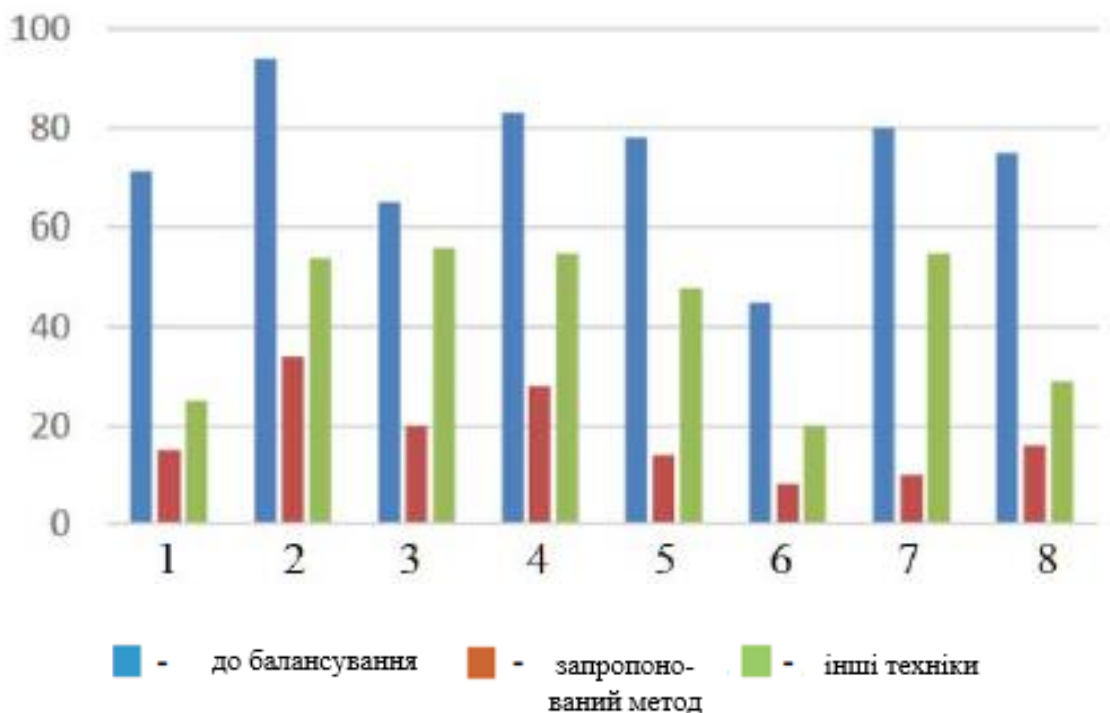


Рисунок 3.13 – Відмінності в показниках робочого навантаження між найбільш завантаженими сайтами та найменш завантаженими сайтами:

тут і далі

На рисунках 3.14 – 3.16 показано вплив використовуваної методики вибору на час відгуку при зміні кількості завдань із 200, 300, 400 до 500. Далі пояснюється значення часу відгуку та визначається формула для його виміри. Результати представлені на рисунках 3.14 – 3.16, відповідають трьом запущеним моделям з різними випадковими робітниками навантаженнями.

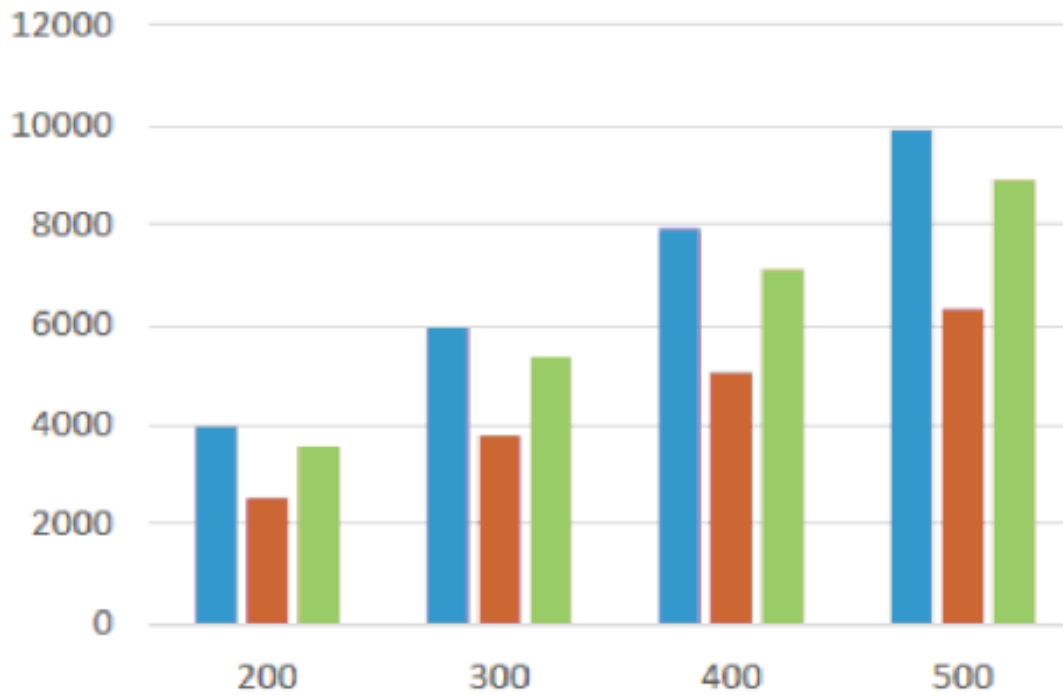


Рисунок 3.14 – Вплив методу вибору на час відгуку в залежності від кількості завдань (набір випадкових робочих навантажень №1)

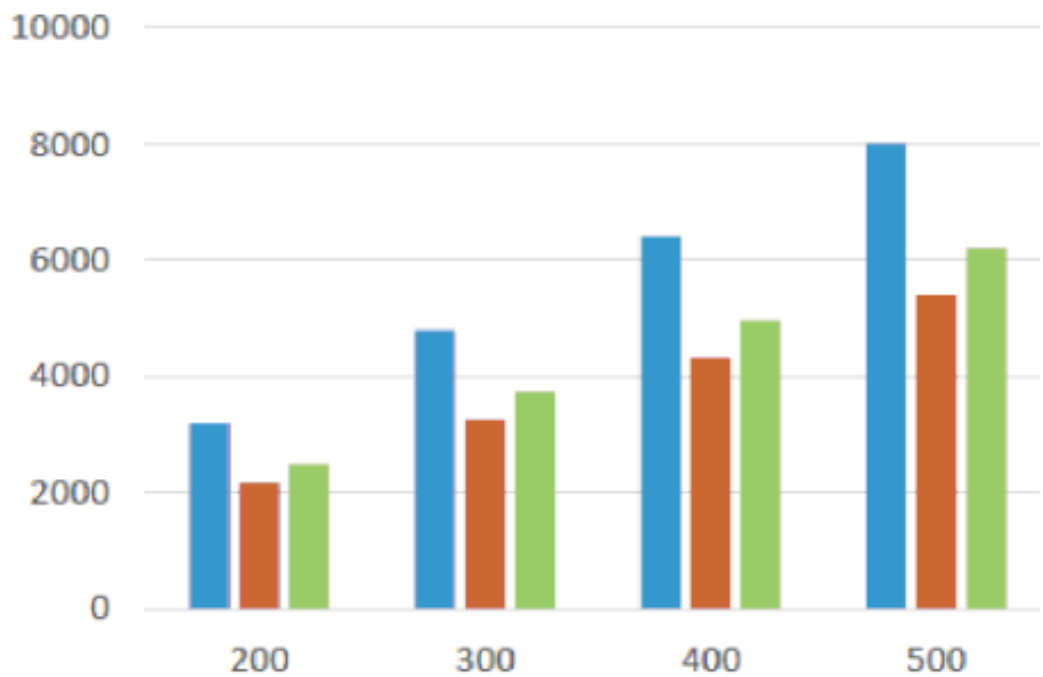


Рисунок 3.15 – Вплив методу вибору на час відгуку в залежності від кількості завдань (набір випадкових робочих навантажень №2)

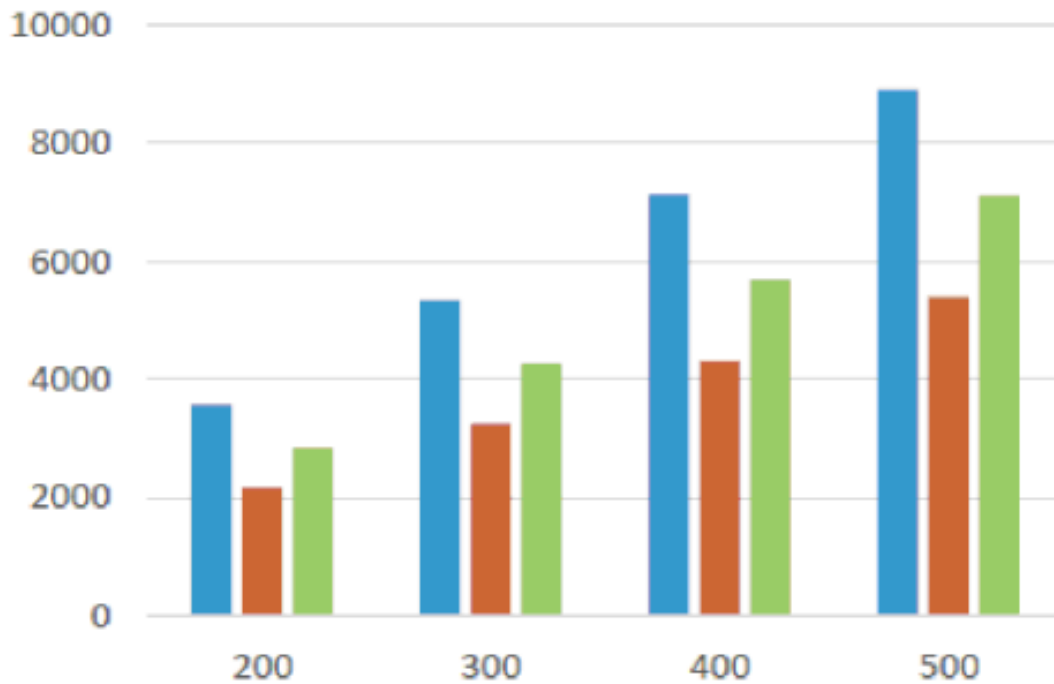


Рисунок 3.16 – Вплив методу вибору на час відгуку в залежності від кількості завдань (набір випадкових робочих навантажень №3)

На рисунках 3.14 – 3.16 показано, що час відгуку збільшується з збільшенням числа завдань, що вимагають виконання більшої кількості робочих навантажень. Крім того, запропонований підхід забезпечує максимально скорочений час відгуку у трьох запущених симуляціях. Це доводить ефективність запропонованої методики вибору при виборі вузла-партнера, і таким чином, всі вузли беруть участь майже з однаковими зусиллями у виконанні завдання, а отже виконання набір завдань відбувається швидше.

### 3.3 Міграція послуг на основі мобільних агентів для забезпечення стійкості до зовнішніх та внутрішніх обурень

Розглянуто проблему міграції послуг на основі мобільних агентів. У системах, які надають послуги у реальному масштабі часу, де показник QoS дуже важливий для користувача, виникають ситуації навантаження платформ, на яких функціонує інформаційне та програмне забезпечення, що

надає послуги.

Підтримка QoS на потрібному рівні вимагає переміщення послуг між платформами, від перевантажених до недовантажених. Для ефективного Розв'язання задачі переміщення необхідний набір методів та інструментів.

Метою роботи є створення методології переміщення у реальному масштаб часу з підтримкою QoS на необхідному рівні. Розділ Містить підхід до переміщення з використанням мобільних агентів.

Важливо, що схема переміщення формується не центром управління, а колективне створення схеми переміщення за допомогою мобільних агентів. Така колективна робота породжує план переміщення, враховуючи параметри надійності, безпеки та завантаженості платформ. Важливою умовою є те, що при переміщенні служб не станеться жодних змін в інших агентів. Фактично таке переміщення агентів зберігає як параметри QoS, а й забезпечує підвищення стійкості до відмови всієї системи. Таким Таким чином, розділ містить підхід до переміщення з використанням мобільних агентів у реальному масштабі часу з підтримкою QoS на необхідному рівні, забезпечуючи підвищення відмовостійкості всієї системи.

Дане дослідження показує застосування міграції мобільних агентів для забезпечення доступності послуг. Таким чином, робота зосереджена на тому, як мобільність агентів допомагає динамічно переміщати служби з некоректних платформ на інші коректні платформи для підвищення доступності послуг. Вона доповнює сучасні дослідження в галузі використання мобільних агентів для підвищення стійкості до відмови системи.

На рисунку 3.17 показано основні компоненти архітектури, в якій 1 СУП – сервер управління подіями, 2 СУР – сервер управління ресурсами, 3 МА – мобільний агент, 4 СВА – середовище виконання агента, 5 П – платформа, 6 ВВОЗ - виявлення вторгнень та оцінка збитків, 7 МС – моніторинг системи.

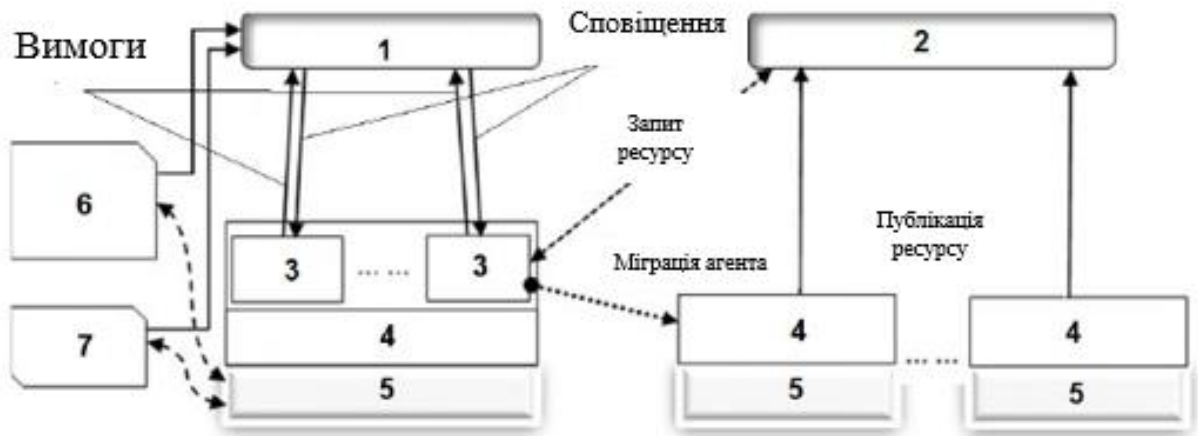


Рисунок 3.17 – Архітектура системи мобільних агентів

Розташування агента в автономному контейнері на платформах, але в кожній часовій точці – на єдиній платформі. RMS та EMS сервери також вирішують завдання обслуговування та допомагають агентам у міжплатформні переміщення. EMS починає процес переміщення, RMS забезпечує агентів ресурсами в рамках контейнерів.

Модель запиту на публікацію використовується для керування ресурсами службою управління правами. У той час, як кожна платформа публікує набір ресурсів (поряд з якістю їх обслуговування), які вона може запропонувати через службу управління правами, агент запитує ресурси та вирішує, чи може платформа підтримувати свої операції. На відміну від моделі push-повідомлень про реєстрацію EMS, запит через службу управління правами призначений для отримання агентом інформації про ресурси, що надаються платформою. У випадку, якщо на поточній платформі агента було виявлено вторгнення в систему безпеки чи платформа не може забезпечити агенту задовільний рівень сервісу, агент повинен запросити служби управління правами, щоб визначити можливі платформи для міграції.

У цьому розділі обговорюються деталі схеми міграції послуг на основі мобільних агентів. Міграція мобільного агента починається з повідомлення EMS про серйозне пошкодження платформи, на якій у В даний час працює агент, або виявлення того, що поточна платформа не може надавати бажаних

послуг. Потім агент повинен вирішити, які можливі платформи слід перейти. Враховуючи взаємодія та функціональна залежність між декількома агентами, які працюють над одним і тим же завданням або групі агентів, які співпрацюють один з одним, необхідно скласти план спільно ї міграції. У наступних розділах обговорюються прийняття рішень про міграцію мобільних агентів та диспетчеризацію агентів.

### 3.4 Прийняття рішень про міграцію локальних та спільних агентів

Рішення агента про перехід на іншу, більш безпечну та надійну платформу ґрунтується на двох важливих факторах. Перший - це відчуття агентом необхідності міграції. Якщо поточна платформа була серйозно пошкоджена та атака може призвести до руйнування інших ресурсів на платформі, агент повинен перейти на іншу платформу. Таким чином, агент може мінімізувати будь-які потенційні втрати та постійно надавати користувачам високоякісні послуги.

Ще одним важливим фактором для ухвалення рішення про перенесення агента є доступність платформ, які можуть надати необхідні агенту ресурси. Агент шукає та ідентифікує ті платформи (через службу управління правами), які задовольняють його функціональним вимогам та вимогам до якості. Потім він вирішує, на які платформи може перейти. Це називається локальним рішенням про міграцію, оскільки рішення приймається одним агентом на основі його локальної інформації. Тим не менш, ресурси, необхідні агентом, повинні вказувати типи ресурсів та відповідні вимоги до якості. Результатом рішення про локальну міграцію є список упорядкованих платформ (з погляду бажаних переваг), на які агент може перейти.

Після того, як кожен окремий агент ухвалить рішення про локальну міграцію, група агентів ухвалить спільне рішення про міграцію щодо остаточного плану міграції. Групове рішення необхідне через взаємозалежність, співробітництво і навіть відносин виключення між

одноранговими агентами в групі для надання послуг. Наприклад, один агент може функціонально залежати від інших агентів (ставлення "залежить від").

Групі агентів може знадобитися працювати на одній платформі (зв'язок атомарності) з таких причин, як спільне використання спільних ресурсів, тісна взаємодія один з одним, виконання набору групових дій або виконання певних спільних операцій із безпеки та конфіденційності. Деякі агенти навіть мають відносини взаємного виключення з таких причин, як розподіл обов'язків, функціональна та соціальна ізоляція або просто організаційна політика; тому вони не повинні мігрувати на ту саму платформу. У запропонованій схемі прийняття рішень ці та інші види операційних відносин агентів представлені у вигляді набору обмежень, які не повинні порушуватись остаточною планом міграції для групи агентів. Спільне прийняття рішень про міграцію полягає у визначенні такого остаточною плану міграції для кожного агента групи.

Рішення про переміщення може бути абсолютно децентралізованим за участю зацікавлених агентів. Разом з тим, оскільки система прийняття рішень працює як система реального часу, у разі виникнення недетермінованих станів системи прийняття рішень може бути проведене не в темпі процесу, а повільніше, і тоді досконала міграція знизить рівень якості обслуговування. Таким чином, доцільно використовувати гібридно-централізований механізм прийняття рішень з їх поєднанням та інтеграцією силами спеціального агента-координатора. На локальному рівні (без втручання спеціального агента-координатора) рішення про локальне переміщення приймається лише локальними агентами. І лише після цього виділяється спеціальний агент-координатор, який приймає від усіх локальних мобільних агентів групи та реєстр платформ (рисунок 3.18).

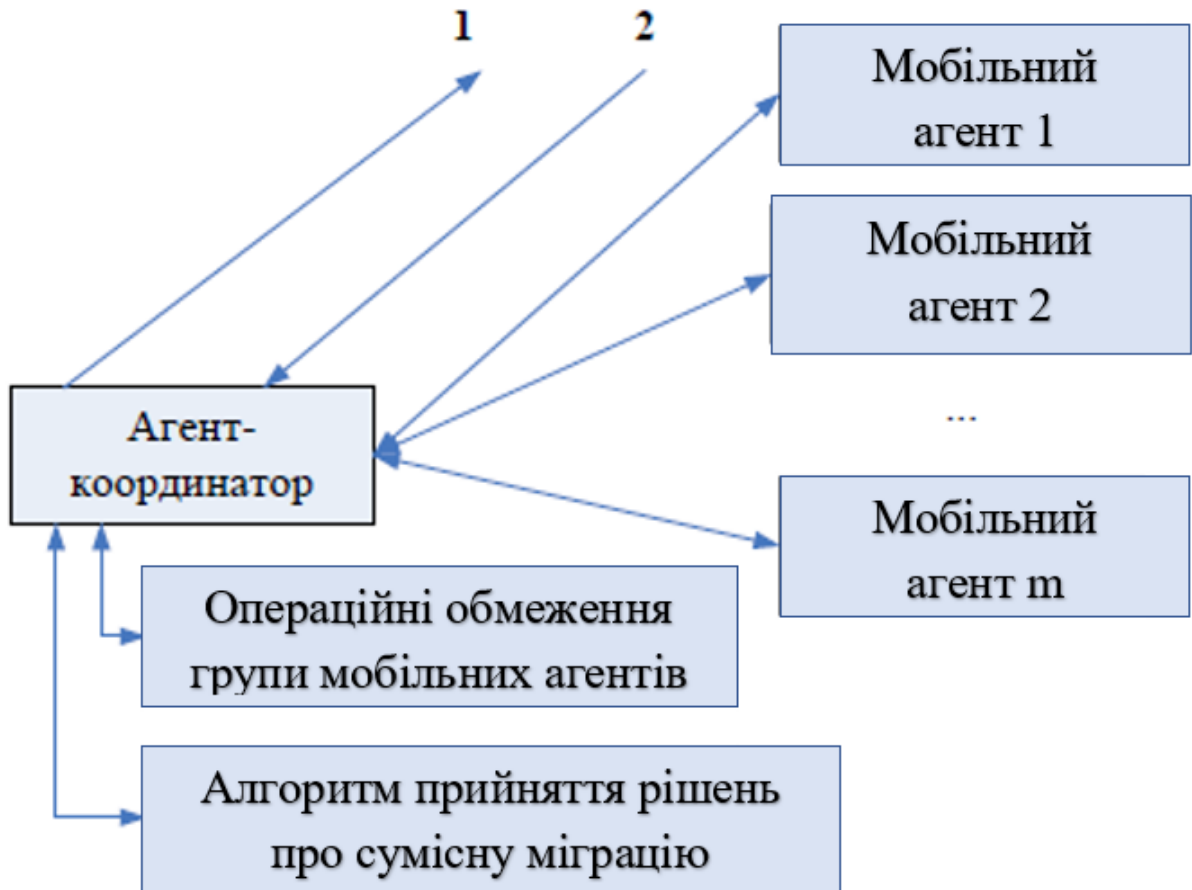


Рисунок 3.18 – Прийняття рішень про спільну міграцію мобільного агента з агентом-координатором: 1 – локальні відомості, вимоги та обмеження міграції (ID агента, множина платформ для міграції);  
2 – кінцевий план міграції

Далі спеціальний агент-координатор враховує відомі обмеження мобільних агентів і намагається створити спосіб групової міграції, що підходить для всієї групи та мобільних агентів, що входять до неї.

Розроблено алгоритм для спільного прийняття рішень про міграцію, який виконує агент-координатор. Алгоритм приймає як вхідні дані рішення про локальну міграцію  $\{S_1, \dots, S_i, \dots, S_m\}$  від усіх  $m$  агентів у групі та набір обмежень  $R$  для групи агентів.

Результатом є остаточний план міграції агентів, якщо такий існує, або повідомлення, що вказує на те, що такий план неможливий, враховуючи

рішення про локальну міграцію поточних агентів та обмеження роботи групи. У  $R$  вказано три типи обмежень [9]. Перше правило атомарності – набір агентів має бути перенесений на ту саму платформу, друге правило залежності – агент функціонально залежить від принаймні від одного з кількох агентів; отже, він повинен мігрувати на ту ж платформу, що і агент, від якого він залежить і третє правило виключення – два або більше агентів не повинні бути перенесені на ту саму платформу. По суті, алгоритм прийняття рішення про спільної міграції рекурсивно вибирає одну платформу з кожного списку можливих платформ, що надаються одним агентом, називається попереднім планом міграції, а потім перевіряє цей попередній план за кожним правилом у списку обмежень. Якщо якийсь обмеження порушено, цей попередній план нездійснений, і алгоритм переходить до наступного попереднього плану міграції доти, доки не будуть вичерпані всі можливі попередні плани або не буде визначено остаточний можливий план міграції. В останньому випадку остаточний план міграції відправляється кожному агенту, щоб він міг розпочати міграцію.

Алгоритм. Спільне ухвалення рішень про міграцію.

Вхід: 1 рішення про локальну міграцію  $S_i$  ( $i=1\dots m$ ) від кожного агента  $A_i$  ( $i=1\dots m$ ), де  $S_i$  містить набір платформ, на які  $A_i$  може мігрувати, а  $m$  представляє кількість мобільних агентів групи;

2 обмеження для роботи групи агентів  $R$ ;

Вихід: План міграції для кожного агента  $A_i$  для переходу на платформу  $P_j$  ( $j = 1 \dots n$ ), де  $n$  представляє кількість доступних платформ.

Початок.

1 Визначити попередній план міграції  $WP$  з одного платформою від кожного  $S_i$ . Спочатку  $WP = \emptyset$ .

2 Для кожного набору платформи  $S_i$ , надісланого агентом  $A_i$  ( $i=1\dots m$ ), виконати:

вибрати платформу  $P_i \in S_i$  і додати  $P_i$   $WP$ , тобто,

$WP = WP \cup \{P_i\}$ .

3 Для кожного обмеження  $r \in R$  виконати якщо є обмеженням залежності, тобто агент  $ag$  залежить принаймні від одного агента в наборі агентів  $A'$ , виконати якщо  $\forall ag' \in A'$ , відповідні платформи  $ag$  та  $ag'$  у  $WP$  різні, виконати  $Reset\ WP = \emptyset$ .

Повернення до кроку 2 для пошуку наступного попереднього плану міграції. Якщо є атомарним обмеженням, тобто. усі агенти повинні бути перенесені на ту саму платформу, виконати.

4 Якщо попередній план міграції  $WP$  задовольняє всім правилам обмежень у  $R$ , повідомити агента  $A_i$  про платформу  $P_i$  для міграції та конкретному розкладі міграції.

5 Інакше оновити кожен  $S_i$  новими ресурсами в системі, які можуть використовувати агент  $A_i$ .

Повернутися до кроку 1 для наступного раунду прийняття рішень про міграції з новими виділеними ресурсами. Закінчення алгоритму.

Якщо у першому раунді алгоритму спільного планування міграції неможливо визначити можливий план міграції, група агентів може перезапустити процес, якщо доступно більше системних ресурсів (крок 5 алгоритму). Агенти перевіряють у службі управління правами, чи доступні деякі нові платформи (деякі сервери резервного копіювання, можливо, були запущені в плану реагування на системні інциденти) або доступні деякі нові ресурси (наприклад, доступні додаткові ресурси системних компонентів чи служб). Цей процес може повторюватися кілька разів доти, доки не буде складено план спільної міграції або не буде досягнуто альтернативної домовленості про переміщення деяких агентів з пошкоджених платформ. У певних ситуаціях мобільним агентам може знадобитися проявити гнучкість, перенастроюючи деякі функції і, таким чином, послабивши певні правила обмежень. Як альтернатива, послуги користувача можуть тимчасово пропонуватися на мінімальному, але прийнятному рівні. У цих випадках може бути можливий спільний план міграції з урахуванням додавання ресурсів чи пом'якшених правил обмежень.

Диспетчеризація агентів. Після складання остаточного плану міграції агентів кожен мобільний агент отримує повідомлення про платформу, на яку він переходить, а також про конкретний розклад міграції, наприклад про порядок міграції. Наприклад, якщо агент A1 функціонально залежить від агента A2, то A2 необхідно спочатку перенести на цільову платформу до A1, щоб A2 міг спочатку відновити свої операції на цільовій платформі. Якщо кілька агентів мігрують з однієї і тієї ж вихідної платформи, необхідне впорядковане відправлення. З іншого боку, якщо агенти з декількох платформ переміщуються на ту саму платформу, їх слід розгортати у визначеному порядку, який враховує їх терміновість та функціональну залежність від інших агентів на тій же платформі.

Технічно відправка агента на нову платформу включає в себе відключення поточних запитів користувачів на обслуговування агента та перенаправлення їх на нову платформу, збереження станів агента, упаковку програм агента та передачу коду агента на нову платформу. У залежності від сховища даних, схемою міграції агента може потрібно також перемістити простір даних агента на нову платформу.

Однак, якщо дані агента зберігаються віддалено, а доступ до даних здійснюється через мережеве підключення (наприклад, сховище даних в хмарі), йому необхідно лише зберегти поточний статус доступу до даних, а потім агент безперервно отримує доступ до даних з новою платформи. В останньому випадку, оскільки немає потреби фізично переміщати дані, це значно знижує навантаження на мережу для міграції агентів і, отже, прискорює весь процес міграції мобільних агентів.

Диспетчеризація мобільних агентів забезпечується API- інтерфейсами Aglets. Складні та докладні операції переміщення агентів (серіалізація об'єктів, синхронізація та переміщення даних) виконуються у середовищі виконання Aglets. Як приклад моделювання спочатку 11 агентів (включаючи EMS та RMS на одному агенті) працювали на платформі.



Рисунок 3.19 – Загальний вид моделювання системи управління 10 мобільними агентами, перенесеними на 2 платформи на головному комп'ютері

Для підтримки міграції агентів у системі були доступні 4 нові платформи. Після запуску наших програм моделювання 10 мобільних агентів були відправлені на відповідні їм нові платформи, яких 8 агентів переїхали на одну платформу, а 2 інших агенти переїхали на іншу платформу (як показано у нижній частині рисунка 3.10). Aglets також забезпечує операцію повернення, яка дозволяє відправленим агентам повернутися на свої вихідні платформи. Переміщення корисне для балансування навантаження системи (для відновлення служб агентів до їхнього початкового розподілу робочого навантаження) та оптимізації обслуговування (щоб переконатися, що користувальницькі програми знаходяться ближче до агентів або щоб переконатися, що програми агентів знаходяться ближче до даних, які вони використовують). Хоча система перевірки концепції не призначена для використання у виробничому додатку, вона демонструє можливість та ефективність переміщення агентів з однієї платформи на іншу за допомогою платформи мобільних агентів, таких як Aglets.

## ВИСНОВКИ

Управління навантаженням є одним із ключових факторів підвищення продуктивності системи. Його основними цілями є збільшення часу виконання завдань та максимальне використання системних ресурсів. У роботі досліджено використання технології мобільних агентів у сфері балансування навантаження. Однією з важливих мотивацій цієї парадигми є мінімізація віддаленого зв'язку, що дозволяє економити споживання смуги пропускання, що сприятливо для ефективної системи балансування навантаження. З цієї причини в запропоноване рішення інтегрований мобільний агент, роль якого полягає у збиранні інформації про навантаження для побудови глобального бачення навантаження на систему, що знижує витрати на зв'язок у порівнянні з класичним способом збору. Крім того, стаціонарні агенти повинні знати про глобальне навантаження, щоб вони могли визначати стан баланс машини і, таким чином, вибирати завдання для перенесення. У запропоноване рішення мобільний наглядний агент інформує тільки перевантажені хости, що призводить до скорочення трафіку по порівняно з тими підходами, які інформують усі хости системи.

Іншими перевагами мобільних агентів є надійність та відмовостійкість, які необхідні в системі балансування навантаження, щоб вона могла продовжувати працювати, коли один із учасників вимкнено. Мобільні агенти також мають перевагу масштабованості, вони добре адаптуються як до невеликих мереж, так та до великомасштабних мереж. Ці переваги дозволяють використовувати мобільні агенти, щоб мати розширювану та надійну систему балансування навантаження.

У великомасштабній мережі збільшення числа контролюючих агентів дозволяє скоротити час міграції агентів та уникнути збільшення розміру коду агента. Це дозволяє покращити балансування навантаження. Порівняльна оцінка результатів показала ефективність підходи до балансування

навантаження. Крім того, показано його вплив на скорочення затримки часу виконання, що є важливим фактором у розподілених системах. Крім того, продемонстровано вплив запропонованих методів відбору порівняно з існуючими методами відбору на балансування робочих навантажень та скорочення часу відгуки системи. Розглянуто проблему міграції послуг на основі мобільних агентів.

Таким чином, представлений підхід до переміщення з використанням мобільних агентів у реальному масштабі часу з підтримкою QoS на необхідному рівні, забезпечуючи підвищення відмовостійкості всієї системи.

Розроблено механізм переміщення мобільних агентів для збереження якості обслуговування або виключення порушення працездатності платформ у разі штатних або нерегламентованих збурень (системні збої, зовнішні атаки, наприклад). У таких ситуаціях мобільні агенти мігруватимуть з повністю або частково втратили працездатність платформ на платформи, що коректно функціонують, для підтримки QoS на необхідному рівні. Розроблено архітектуру системи, визначено процедури міграції агентів та проведено моделювання для запропонованої схеми міграції служб на основі агентів.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Є.В. Момотов, О.О. Можасєв. Управління навантаженням у розподілених системах та парадигма мобільних агентів. Тези доповідей 12-ої міжнародної науково-технічної конференції «Проблеми інформатизації», 21-22 листопада 2024р. Баку: ІСУ – Харків: ХНУРЕ, 2024 – Т.1., С.58.
2. Ahmed YE, Adjallah KH, Stock R, Kacem I, Babiker SF. NDSC based methods for maximizing the lifespan of randomly deployed wireless sensor networks for infrastructures monitoring. *Comput Indust Eng.* 018;115:17-25. <http://www.sciencedirect.com/science/article/pii/S0360835217304667>.
3. Al-Kiyumi RM, Foh CH, Vural S, Chatzimisios P, Tafazolli R. Fuzzy logic-based routing algorithm for lifetime enhancement in heterogeneous wireless sensor networks. *IEEE Trans Green Commun Netw.* 2018;2(2):517-532.
4. Liu X. An optimal-distance-based transmission strategy for lifetime maximization of wireless sensor networks. *IEEE Sens J.* 2015;15(6): 3484-3491.
5. Kale PA, Nene MJ. Path reestablishment in wireless sensor networks. In: 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET); 2017:1659-1663.
6. Sadeghi F, Avokh A. Load-balanced data gathering in internet of things using an energy-aware cuckoo-search algorithm. *International Journal of Communication Systems.* 2020;33(9):e4385. <https://doi.org/10.1002/dac.4385>.
7. Liang W, Xu Y, Shi J, Luo J. Aggregate node placement for maximizing network lifetime in sensor networks. *Wirel Commun Mobile Comput.* 2012;12(3):219-235. <https://doi.org/10.1002/wcm.952>.
8. Jothiprakasam S, Muthial C. A lifetime enhancement protocol in cluster-based wireless sensor networks with guaranteed delay. *Int J Commun Syst.* 2019;32(12):e3983. <https://doi.org/10.1002/dac.3983>.
9. Kale PA, Nene MJ. Scheduling of data aggregation trees using local heuristics to enhance network lifetime in sensor networks. *Computer Networks.*

2019;160:51-64.

10. Liu X, Liu A, Wang T, Ota K, Dong M, Liu Y, Cai Z. Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks. *J Parall Distrib Comput*. 2020;135:140-155.

11. Liu X, Zhao M, Liu A, Wong KKL. Adjusting forwarder nodes and duty cycle using packet aggregation routing for body sensor networks. *Inf Fusion*. 2020;53:183-195.

12. Wang J, Gao Y, Yin X, Li F, Kim H-J. An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks. *Wirel Commun Mobile Comput*. 2018;2018.

13. Wang J, Gao Y, Liu W, Wu W, Lim S-J. An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks. *Comput Mater Contin*. 2019;58:711-725.

14. Wang J, Gu X, Liu W, Sangaiah AK, Kim H-J. An empowerhamilton loop based data collection algorithm with mobile agent for WSNs. *Human-centric Comput Inf Sci*. 2019;9(1):1-14.

15. Wang J, Gao Y, Wang K, Sangaiah AK, Lim S-J. An affinitypropagation-based self-adaptive clustering method for wireless sensor networks. *Sensors*. 2019;19(11):2579.

16. Wang J, Gao Y, Zhou C, Sherratt RS, Wang L. Optimal coverage multi-path scheduling scheme with multiple mobile sinks for wsns. *Comput Mater Continua*. 2020; 62(2):695-711. <http://www.techscience.com/cmc/v62n2/38271>.

17. Zhou F, Chen Z, Guo S, Li J. Maximizing lifetime of data-gatheringtrees with different aggregation modes in WSNs. *IEEE Sens J*. 2016; 16(22):8167-8177.

18. Jifeng H., Li X., Liu Z., *Component-based software engineering// Theoretical Aspects of Computing–ICTAC*, Springer, 2005, p. 70-95.

19. Zhang X. et al. Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing// *Mobile Networks and Applications*, 2011. 16(3), p. 270-284.

20. Gao F., Curry E., Ali M.I., Mileo A. QoS-Aware Complex Event Service Composition and Optimization Using Genetic Algorithms// Lecture Notes in Computer Science, 2014, vol. 8831. DOI: 10.1007/978-3-662-45391-28.