ISSN 2522-9818 (print)
ISSN 2524-2296 (online) *Innovative technologies and scientific solutions for industries. 2023. No. 3 (25)*

УДК 004.75 : 004.451.7 DOI: https://doi.org/10.30837/ITSSI.2023.24.088

Б. Резанов, Г. Кучук

МОДЕЛЬ РОЗПОДІЛУ ЕЛЕМЕНТАРНИХ ПОТОКІВ ДАНИХ У ТУМАННІЙ ПЛАТФОРМІ ПІДТРИМКИ ІНТЕРНЕТУ РЕЧЕЙ

Предметом дослідження є моделі управління ресурсами та завданнями в туманному середовищі підтримки Інтернету речей (IoT). Зростання кількості підключених пристроїв та обсяги зібраної інформації в мережах IoT роблять актуальним необхідність удосконалення систем управління, які забезпечують оптимальний розподіл завдань і ресурсів. Туманне обчислення відіграє ключову роль у реалізації цього питання, розподіляючи обчислювальні завдання ближче до джерела інформації та кінцевих користувачів. **Мета роботи** полягає в підвищенні ефективності технологій туманних обчислень для забезпечення оптимального розподілу завдань і ресурсів у мережі IoT. Для досягнення мети розглянуті методи кластеризації, що допомагають створити групи обчислювальних ресурсів і визначити, які завдання необхідно розподілити між цими групами. Застосування відповідних методів кластеризації дає змогу зменшити затримки та підвищити загальну продуктивність системи IoT. **Основні завдання цієї роботи.** По-перше, зважаючи на різноманітні вимоги до обчислювальних ресурсів та завдань IoT, необхідно розглянути наявні методи й розробки. По-друге, важливо дослідити та порівняти методи кластеризації, зокрема *DBSCAN* та *C-Means*, для ефективного управління ресурсами. Метод кластеризації *DBSCAN* дає змогу ефективно розподіляти завдання залежно від їх місця розташування. Метод *C-Means* допомагає групувати ресурси за їх характеристиками. Третє завдання – розробити модель, основану на вхідних параметрах, таких як відповідь системи, потреби кластерів у ресурсах, віддаленість інформації для її оброблення тощо. Модель дасть змогу аналізувати ймовірні сценарії та приймати рішення щодо оптимального розподілу завдань і ресурсів у середовищі IoT. **Висновки.** Це дослідження спрямоване на розв'язання актуального питання – управління ресурсами та завданнями в туманному середовищі IoT. Розглянуто наявні методи й розробки у сфері управління ресурсами та завданнями в IoT. Порівняно методи кластеризації *DBSCAN* і *C-Means* для визначення їх ефективності в управлінні ресурсами. Розроблено теоретико-множинну модель, що ґрунтується на різних параметрах для прийняття оптимальних рішень щодо розподілу завдань і ресурсів. Установлено, що впровадження методів кластеризації та розробленої моделі допомагають підвищити продуктивність системи й забезпечити більш ефективне застосування обчислювальних ресурсів у туманному середовищі IoT.

Ключові слова: туманні обчислення; Інтернет речей; метод *DBSAN*; метод *C-Means*; кластеризація; теоретико-множинне моделювання.

Вступ

У сучасному світі технології розвиваються на шаленій швидкості, відкриваючи перед людством безмежні можливості та виклики. Інтернет речей (IoT) [1] є однією з найбільш яскравих технологічних революцій, що змінює наше оточення та спосіб життя. Завдяки IoT об'єкти зовнішнього світу, починаючи від побутових приладів до великих виробничих систем, можуть спілкуватися, обмінюватися інформацією та приймати рішення, не потребуючи прямої участі людини.

Проте реалізація IoT та забезпечення його працездатності вимагають ефективного управління ресурсами й завданнями. На допомогу приходять туманні обчислення, які дають змогу обробляти інформацію та приймати рішення на самому "краю" мережі, ближче до джерела їх виникнення [2, 3]. Але як можна забезпечити ефективне управління в умовах туманного середовища?

У цій статті досліджується два ключові аспекти цього питання: методи кластеризації та система розподілу завдань у туманному середовищі підтримки IoT. Методи кластеризації допомагають групувати ресурси та об'єкти в єдиноцілісні структури, що полегшує їх ефективне використання. Система розподілу завдань забезпечує оптимальний розподіл завдань між ресурсами в мережі IoT [4].

Розглянемо метод кластеризації *DBSCAN (Density-Based Spatial Clustering of Noisy Applications)* [5, 6], що дає змогу створювати кластери з ресурсів, ґрунтуючись на їх розташуванні, і метод

Сучасний стан наукових досліджень та технологій в промисловості. 2023. № 3 (25)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

кластеризації *C-Means* [7], розподілений на *CPU Cluster, GPU Cluster* та *RAM Cluster* для ефективного використання обчислювальних ресурсів. Теоретико-множинна модель допоможе зрозуміти, як оптимально розподіляти завдання та ресурси.

Аналіз літератури

Аналіз літератури вказує на декілька ключових проблем, пов'язаних із підтримкою IoT та кластеризацією даних, що розглядаються у статті.

1. Ефективне управління інформацією IoT у хмарних середовищах. У дослідженні [1], наголошується на важливості хмарних технологій для оброблення та зберігання даних IoT. Проте це може призвести до питань щодо ефективного управління ресурсами хмари.

2. Кластеризація та оброблення даних IoT. У дослідженнях [5, 6] висвітлюється важливість алгоритмів кластеризації даних IoT для подальшого аналізу та оброблення. Проте існує проблема в обранні найбільш ефективного алгоритму для конкретних завдань.

3. Енергозбереження в смарт-містах. У статті [4] вказано на необхідність мінімізації споживання енергії датчиками IoT в смарт-містах. Ця проблема стосується оптимізації роботи датчиків та їх енергоефективності.

4. Паралельні обчислення на *GPU*. У роботах [11, 14] описані переваги використання графічних процесорів (*GPU*) для швидкодії та паралельних обчислень. Проте існує проблема оптимального розподілу завдань між різними обчислювальними ресурсами.

5. Розподілені обчислення. У дослідженнях [5, 8–10] наголошується на важливості розподілених обчислень, зокрема на використанні платформи *Spark*. Але питання ефективності та оптимізації розподілених обчислень залишається актуальним.

Мета статті – розглянути способи розв'язання перелічених проблем, зокрема розробити систему розподілу завдань із використанням методів кластеризації та оптимізації ресурсів у туманних середовищах підтримки IoT. Для цього створено теоритико-множинну модель для визначення оптимального способу розподілу завдань і ресурсів. Такий підхід спрямований на покращення продуктивності та ефективності системи підтримки IoT на туманній платформі.

1 Методи кластеризації туманного середовища підтримки IoT

Ключовими складниками успішної системи туманного обчислення є методи кластеризації, що дають змогу раціонально розподіляти завдання та ресурси в системі. Два основних методи кластеризації, що застосовуються в цій роботі, – це *DBSCAN* і *C-Means Clustering*. Розглянемо їх докладніше.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) – метод кластеризації, що оснований на щільності інформації та допомагає виявляти об'єкти, які належать до одного кластера на основі їх географічного розташування [8]. У досліджуваній системі використано *DBSCAN* для організації обчислювальних ресурсів у різних локаціях. Наприклад, *DBSCAN Cluster 1* може відповідати одному регіону, *DBSCAN Cluster 2* – іншому і т.д. [9, 10].

C-Means Clustering, або нечітка кластеризація, застосовується для подальшої кластеризації ресурсів у кожному *DBSCAN Cluster*. Розбито ресурси на кластери, такі як *CPU Cluster, GPU Cluster* і *RAM Cluster*, щоб ефективно розподіляти завдання з огляду на тип ресурсів [11].

Така комбінація методів кластеризації дає змогу оптимізувати оброблення інформації, розподіляти завдання та забезпечувати високу продуктивність в елементарних потоках даних у туманній платформі підтримки IoT. Математичні моделі та алгоритми цих методів допомагають у прийнятті рішень та покращенні функціювання системи.

Побудуємо діаграму кластеризації двома обраними методами (рис. 1).

Ця діаграма відображає систему, що використовує два методи кластеризації для управління завданнями й ресурсами в туманному середовищі підтримки IoT.

Туманне середовище

Система поділена на три кластери *DBSCAN*, позначені як *DBSCAN Cluster 1, DBSCAN Cluster 2* та *DBSCAN Cluster 3*. Кожному кластеру *DBSCAN* відповідають різні локації, де виконуються обчислення. У кожному кластері *DBSCAN* є вузли, позначені як *Node X*, де *X* – це номер вузла.

Кластери C-Means

Усередині кожного кластера *DBSCAN* є підкластери *C-Means*, що подані як *C-Means Cluster*. *CPU Cluster* призначений для оброблення на *CPU*.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2023. No. 3 (25)

RAM Cluster – для оброблення завдань в оперативній пам'яті (*RAM*).

Зв'язки

Вузли кожного кластера *DBSCAN* взаємодіють з відповідними кластерами *C-Means* для оброблення завдань. Наприклад, *Node 1* з *DBSCAN Cluster 1* взаємодіє з *CPU Cluster* для оброблення завдань на *CPU*.

Також відображено зв'язки між кластерами *DBSCAN* і *C-Means*, що показує, як ресурси та завдання розподіляються між різними локаціями та видами обчислень [12].

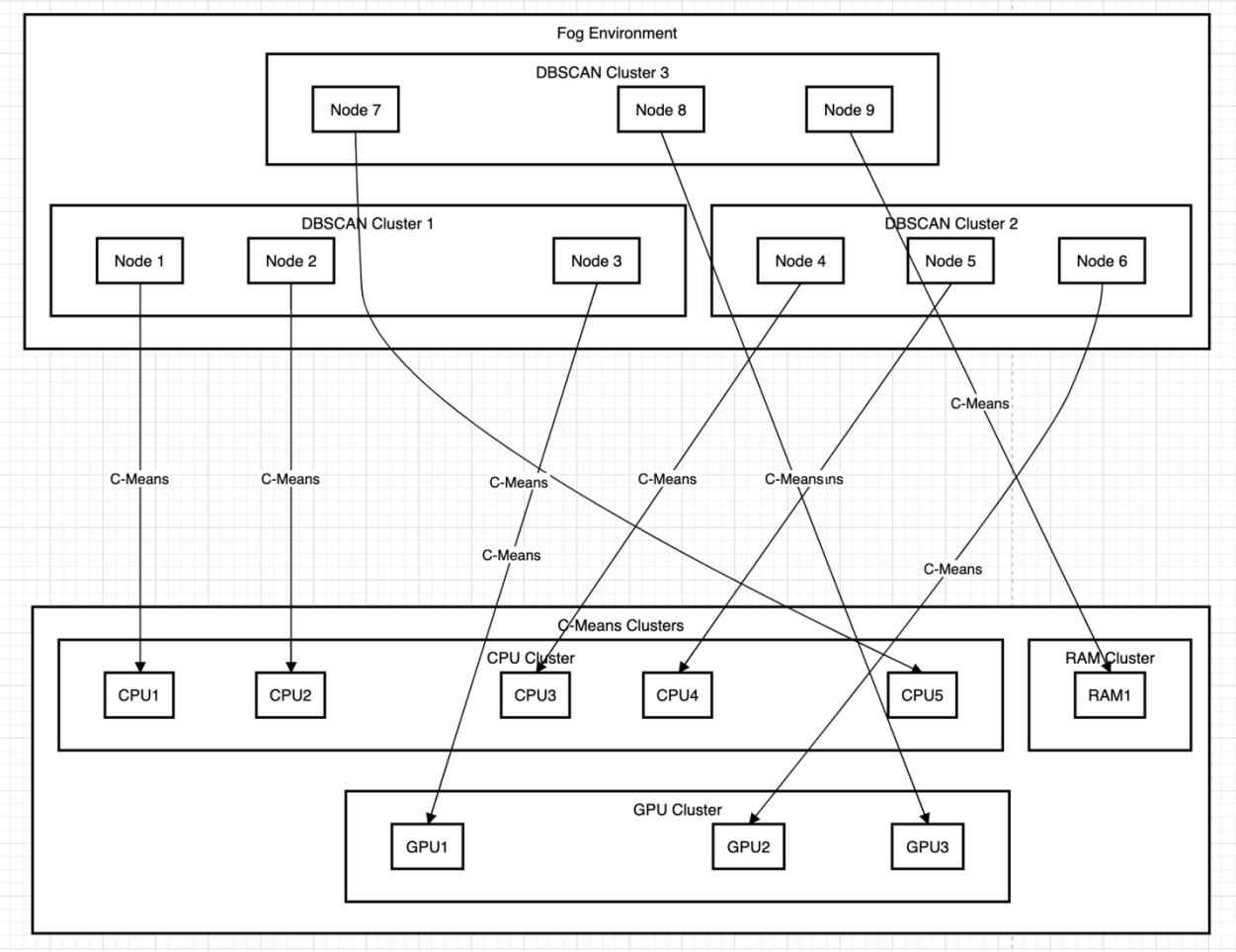

Рис. 1. Діаграма кластеризації двома методами *DBSCAN* і *C-Means Clustering*

Наведена система дає змогу ефективно управляти завданнями та ресурсами в розподіленому середовищі обчислень туманної платформи підтримки IoT, використовуючи два методи кластеризації для оптимізації оброблення даних у реальному часі.

Після розгляду методів кластеризації та їх застосування в системі туманного обчислення опишемо систему розподілу завдань і ресурсів у туманному середовищі. Діаграма й методи кластеризації визначають ключові аспекти організації роботи системи [13]. Варто наголосити, що ці методи кластеризації мають важливе значення не лише для туманного обчислення, але й для багатьох галузей науки та промисловості. Кластеризація дає змогу групувати схожі об'єкти для подальшого аналізу, оброблення та прийняття рішень.

Методи кластеризації стали основою для створення ефективної системи розподілу завдань і ресурсів у мережі туманної платформи. Вони допомагають ідентифікувати оптимальні рішення для оброблення інформації на краях мережі, забезпечуючи найкращий розподіл завдань між кластерами та краями з огляду на фізичне розташування, доступні ресурси та час відповіді.

Сучасний стан наукових досліджень та технологій в промисловості. 2023. № 3 (25)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

У наступних розділах більш детально розглянемо теоретико-множинну модель системи, що допомагає оптимізувати процеси розподілу завдань та кластеризації ресурсів, зробивши платформу туманного обчислення більш продуктивною та відповідною до вимог сучасних застосунків Інтернету речей та інших сфер застосування

2 Система розподілу завдань у середовищі туманних обчислень, підтримки IoT

Система розподілу завдань у середовищі туманних обчислень, підтримки IoT покладається на кілька вхідних параметрів, зокрема затримку відповіді, ємність кластера та віддаленість інформації для оброблення [14]. Система використовує два типи методів кластеризації туманного середовища: *C-Means* (з використанням кластерів *CPU, GPU* та *RAM*) і *DBSCAN* (на основі розташування кластерів 1, 2 і 3). Завдання зберігаються в *S3*-подібному сховищі, особливому для кожного кластера *DBSCAN*. Зображення системи подане на рис. 2.

Затримка відповіді (у секундах). Цей параметр визначає максимально допустиму затримку для виконання завдань й гарантує, що вони обробляються протягом визначеного часу.

Ємність кластера (у довільних одиницях). Ємність кластера вказує на обчислювальні ресурси, доступні в кожному кластері, зокрема *CPU, GPU* та *RAM* [15]. Цей параметр кількісно визначає максимальне робоче навантаження, що може ефективно впоратися з кластером.

Віддаленість інформації для оброблення. Віддаленість даних відображає відстань або затримку мережі між джерелом інформації й туманом, призначеним для оброблення. Дуже важливо звести до мінімуму затримки передачі даних.

Генерація завдань. Пристрої IoT створюють різні обчислювальні завдання, кожне зі своєю складністю та рівнем пріоритету.

Профілювання завдань. Кожне завдання профільовано, щоб визначити його конкретні вимоги, зокрема обмеження часу відповіді, вимоги до обчислювальних ресурсів і інформацію про географічне розташування.

Кластеризація крапель туману. Система використовує метод кластеризації для організації крапель туману на основі їх географічного розташування. Краплі туману організовані в кластери, що відповідають локації 1, локації 2 або локації 3.

Призначення кластера. Завдання спочатку розподіляються за місцем, у якому вони були створені. Система оцінює доступність ресурсів у місці створення завдання та перевіряє обмеження часу відповіді. Якщо ресурсів не достатньо або обмеження часу відповіді не може бути виконане в місці походження завдання, система шукає альтернативні місця з доступними ресурсами та меншою затримкою.

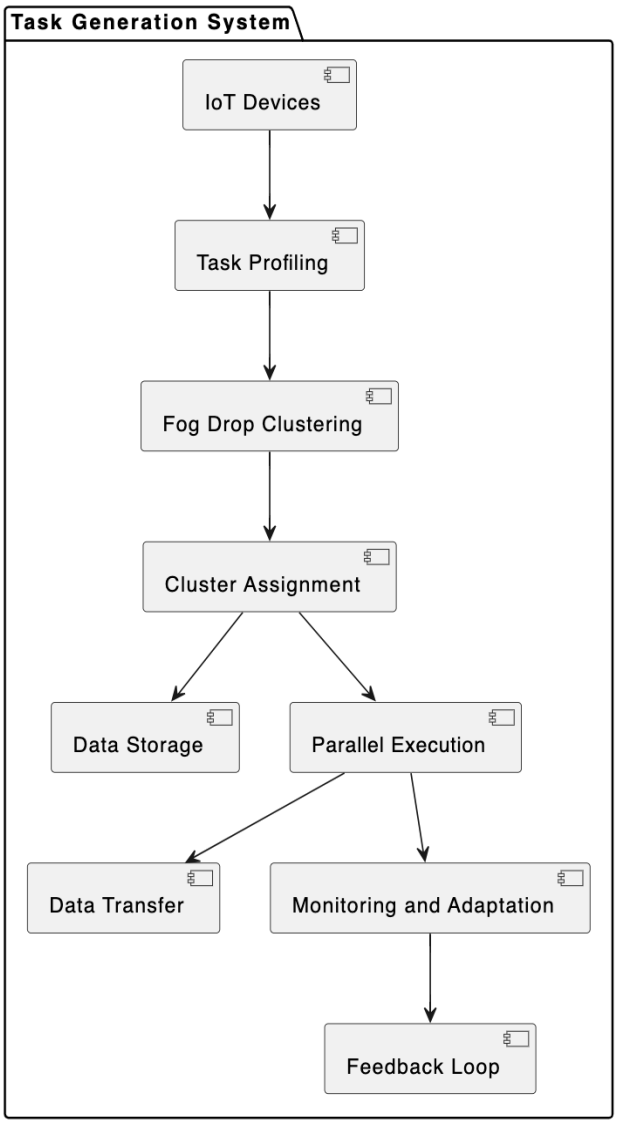

Рис. 2. Опис системи розподілу завдань і ресурсів

На основі цієї оцінки система перепризначає завдання кластеру, який найкраще відповідає її вимогам. Наприклад, якщо для завдання потрібне оброблення *GPU*, а ресурси в розташуванні

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2023. No. 3 (25)

локації 1, але в локації 3 є доступні ресурси *GPU* з меншою затримкою порівняно з розташуванням 2, завдання перенаправляється в локацію 3, зокрема в кластер *GPU* [16].

Зберігання даних. Завдання зберігаються у спеціальному *S3*-подібному сховищі в кожному кластері, що забезпечує доступність інформації для оброблення.

Паралельне виконання. У кожному кластері завдання виконуються паралельно, ефективно використовуючи доступні ресурси, дотримуючись обмежень часу відповіді.

Передача даних. Якщо завдання потребує інформації, якої наразі немає у призначеному кластері, запускаються механізми передачі даних, щоб перемістити необхідну інформацію ближче до вузла оброблення й мінімізувати віддаленість даних.

Моніторинг і адаптація. Система постійно контролює виконання завдань, навантаження на кластер і час відповіді. Якщо кластер наближається до обмежень ємності або час відповіді перевищує вказані порогові значення, система може перерозподілити завдання іншим кластерам або за потреби виділити додаткові ресурси.

Цикл зворотного зв'язку. Показники продуктивності, зокрема час відповіді та використання ресурсів, збираються та аналізуються. Ці показники слугують зворотним зв'язком для вдосконалення стратегій розподілу завдань, адаптації до змінних умов і оптимізації розподілу завдань у часі.

Беручи до уваги перелічені вхідні параметри та використовуючи методи кластеризації крапель туману, середовище підтримки IoT ефективно обробляє завдання, забезпечуючи вчасне виконання, ефективне застосування ресурсів і мінімізовану віддаленість інформації на основі географічного розташування. Цей підхід покращує керування застосунками Інтернету речей у середовищі туманних обчислень.

Методи розподілу завдань і кластеризації ресурсів є критично важливими складниками для досягнення оптимального функціювання системи підтримки Інтернету речей в умовах туманного обчислення. У змінному та динамічному середовищі, де багато факторів може впливати на прийняття рішень, застосування теоретико-множинних моделей стає невід'ємним складником для покращення цих процесів.

3 Теоретико-множинна модель

Теоретико-множинна модель є відображенням системи, якій властиві важливі параметри та їх взаємозв'язки. У нашій конкретній ситуації теоретико-множинна модель містить функціональні залежності між вхідними параметрами, такими як відповідь системи на завдання, потреби кластерів у ресурсах і віддаленість інформації, яка потребує оброблення [17].

Було створено теоретико-множинну модель, що дає змогу системі аналізувати ймовірні сценарії та приймати найефективніші рішення щодо розподілу завдань з метою максимізувати продуктивність і забезпечити вчасне виконання завдань. Використання теоретико-множинних моделей допомагає системі оптимізувати роботу, забезпечуючи краще задоволення потреб користувачів і максимальну результативність.

3.1 Потік робіт

Потік робіт може бути поданий у вигляді орієнтованого ациклічного графа *(ОАГ)*:

$$H = (V, E), \quad (1)$$

де H – монолітний потік робіт;

V – множина вершин, що становлять обчислювальні завдання;

E – множина спрямованих ребер, які з'єднують вершини, що становлять залежності за показниками між завданнями.

Кожна вершина v_i в V може мати вхідні та/або вихідні ребра. Вихідний ступінь $\deg^+(v_i)$ вершини v_i в H визначається як кількість ребер, що виходять з v_i, спрямованих до інших вершин. Вхідний ступінь $\deg^-(v_i)$ вершини v_i в H визначається як кількість ребер, спрямованих до v_i від інших вершин. Ребро $(v_i, v_j) \in$ є залежністю за даними від v_i до v_j. Нехай n – це загальна кількість вершин в V.

3.2 Підпотоки робіт

Потік робіт H поділяється на множину, що складається з k підпотоків робіт $S = (S_1, \ldots S_k)$, $S_i = (V_i, E_i)$, якщо V_i – це множина вершин, що

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2023. № 3 (25) ISSN 2524-2296 (online)

входять у підпотік робіт S_i; E_i – це множина ребер між вершинами, що входять до V_i. У цьому разі множина вершин H поділяється підпотоками робіт на набір непересічних підмножин:

1) $\forall i \in 1..k: V_i \subset V, E_i \subset E;$

2) $\forall v \in V, \exists i \in 1..k : v \in V_i;$

3) $E_i = \{(v_k, v_l) \in E : v_k, v_l \in V_i\};$

4) $\forall i, j : i \neq j \Rightarrow V_i \cap V_j = \varnothing.$

Визначимо наступні класи ребер і вершин, пов'язаних із підпотоком робіт S_i:

1) EI_i: набір вхідних ребер із початковою вершиною поза підпотоком робіт S_i і кінцевою вершиною всередині підпотоку робіт S_i:

$$EI_i = \{(v_k, v_l) \in E : v_k \notin S_i, v_l \in S_i\}; \quad (2)$$

2) EO_i: набір вихідних ребер із початковою вершиною всередині підпотоку робіт S_i і кінцевою вершиною поза підпотоком робіт S_i:

$$EO_i = \{(v_k, v_l) \in E : v_k \in S_i, v_l \notin S_i\}; \quad (3)$$

3) VI_i: набір вершин в S_i, розташованих на основній частині ребер EI_i, а також вершин, що не мають вхідних ребер:

$$VI_i = \{v_l \in S_i : (v_k, v_l) \in EI_i\} \cup \{v \in S_i : \deg^-(v) = 0\}; \quad (4)$$

4) VO_i: набір вершин в S_i, розміщених на кінцях ребер EO_i, а також вершин, що не мають вихідних ребер:

$$VO_i = \{v_k \in S_i : (v_k, v_l) \in EO_i\} \cup \{v \in S_i : \deg^+(v) = 0\}. \quad (5)$$

3.3 Визначення елементарного потоку робіт

Щоб перетворити підпотік робіт S_i в елементарний потік робіт (*Elemental Flow*), необхідно витягти всі вершини S_i з потоку робіт H і забезпечити комунікаційні механізми, що пов'язують EF_i з платформою потокової передачі подій через вершину-споживач cv_i та вершину-генератор pv_i.

Вершина cv_i в EF_i забезпечує споживання потоку вхідних даних від платформи потокової передачі даних і розподіляє його між вершинами VI_i. Вершина pv_i в EF_i діє як сток, що збирає вихідні дані з вершин, що становлять безліч VO_i і передає їх у вигляді повідомлень на платформу потокової передачі даних. Визначимо відповідні множини ребер таким чином:

$ECV_i = \{(cv_i, v) : v \in VI_i\}$, набір ребер, що йдуть від cv_i до вершин у VI_i;

$EPV_i = \{(v, cp_i) : v \in VO_i\}$, набір ребер, що йдуть від вершин у VO_i до cp_i.

У цьому разі елементарний потік робіт EF_i з підпотоку робіт S_i визначається як

$$EF_i = (TV_i, TE_i), \quad (6)$$

де $TV_i = V_i \cup \{cv_i, pv_i\}$ – множина вершин, що розташовані всередині S_i, зокрема cv_i та pv_i;

$TE_i = E_i \cup ECV_i \cup EPV_i$ – множина всіх ребер, розташованих усередині S_i, зокрема всі ребра, які йдуть із cv_i до вершин у VI_i, а також всі ребра, що йдуть від вершин у VO_i до cp_i.

3.4 Модель елементарних потоків робіт

Модель елементарних потоків робіт поєднує моделі потоків робіт і потокового оброблення інформації. У структурі платформи потокового оброблення інформації формується набір виділених каналів (сховищ потоків даних) для організації взаємодії елементарних потоків робіт за допомогою обміну повідомленнями. Кожне повідомлення є набором даних, що містить:

- познаку часу створення повідомлення;
- інформацію про джерело даних;
- структуровану колекцію даних, що передаються.

DS_0 відповідає за збирання, зберігання та надання повідомлень, що містять набори даних, необхідних для ініціалізації обчислювального процесу в потоці робіт;

DS_i відповідає за отримання повідомлень, що містять проміжні дані, від EF_i, і передачу їх залежним елементарним потокам робіт;

DS_{out} відповідає за збирання, зберігання та надання повідомлень, що містять результуючі дані.

Замість початкового вузла потоку робіт, набори даних, необхідні для початку обчислювального

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2023. No. 3 (25)

процесу, надходять у сховище потоків даних DS_0 у вигляді повідомлень. Оброблення повідомлень зі сховища потоку даних організоване таким чином:

1) cv_i відповідного EF_i витягує наступне повідомлення з DS_{i-1};

2) на основі аналізу отриманого повідомлення cv_i ініціює передачу даних по ребрах ECV_i до вершин, відповідальних за виконання безпосереднього обчислення;

3) після завершення завдань оброблення даних та їх надсилання по ребрах EPV_i pv_i генерує вихідне повідомлення в DS_i або DS_{out}, якщо це остаточний результат.

На рис. 3 зображено застосування моделі елементарних потоків робіт. Блок $EF1$ є елементарним потоком робіт і містить вершину системи прийняття рішення cv та вершину відповідача pv, обчислювальні завдання V_i. Блок $EF1$ після виконання своїх завдань передає інформацію в сховище DS_1 для подальшого оброблення. Блок $EF2$ є аналогом блоку $EF1$, але зі своїми обчислювальними завданнями V_i. Стрілками показано переміщення інформації з системи прийняття рішень до виконання безпосереднього обчислювального процесу. Пунктирними стрілками показано переміщення інформації в/із сховища потоків даних. Унаслідок застосування запропонованої моделі було отримано можливість відокремити елементарні потоки даних із загальної інформації (підпотоків даних). Відокремлення елементарних потоків даних дало змогу покращити якість оброблення й аналізу інформації та швидкість оброблення даних.

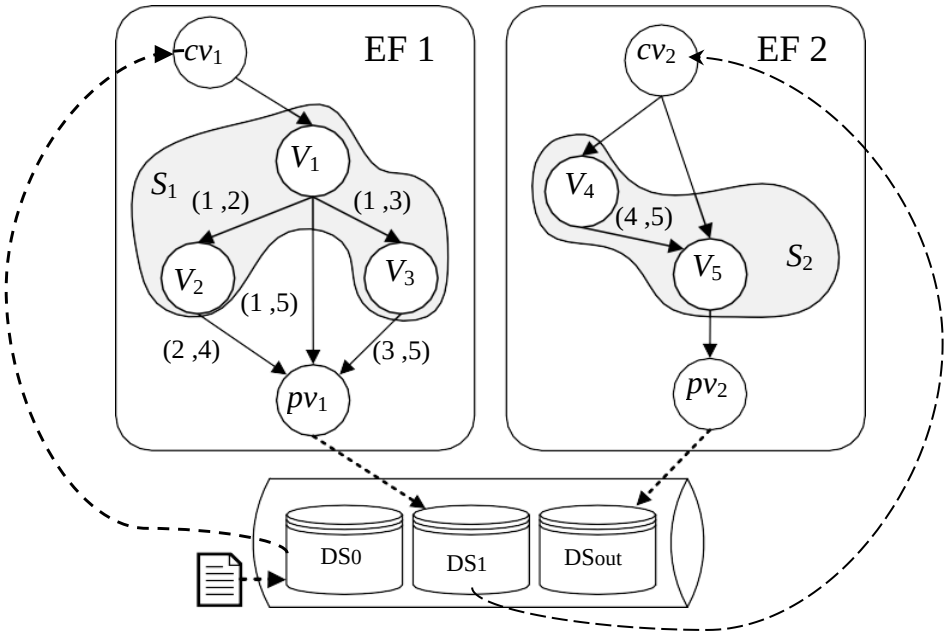

Рис. 3. Результат застосування моделі елементарного потоку робіт

Запропонована модель враховує параметри, зокрема пропускну спроможність мережі, підключення між різними елементами системи та більшу кількість змінних для оптимізації розподілу елементарних потоків даних у туманному середовищі підтримки IoT.

Висновок

У роботі досліджено та впроваджено два ключові методи кластеризації, що виявилися надзвичайно важливими для оптимізації та управління завданнями й ресурсами в туманному середовищі підтримки Інтернету речей (IoT). Подана діаграма ілюструє важливий етап у розвитку системи, яка може істотно покращити продуктивність та ефективність використання ресурсів у мережах туманної платформи підтримки IoT.

Метод *DBSCAN*, оснований на локаціях кластерів, і метод *C-Means*, розподілений на *CPU Cluster*, *GPU Cluster* та *RAM Cluster*,

Сучасний стан наукових досліджень та технологій в промисловості. 2023. № 3 (25)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

дають змогу системі ефективно визначати, куди та як розподіляти завдання й ресурси. Ці методи допомагають уникати перевантаження й забезпечують розподіл ресурсів відповідно до вимог завдань.

Запропонована теоретико-множинна модель – необхідний інструмент для аналізу розподілу завдань і кластеризації ресурсів. Вона дає змогу передбачити рішення, забезпечуючи максимальну продуктивність та вчасне виконання завдань.

Розглянута система розподілу завдань і ресурсів має значний потенціал для застосування в різних сферах, де важлива ефективність роботи IoT і туманного обчислення. Вона може допомогти вирішувати завдання в реальному часі, оптимізувати використання обчислювальних ресурсів і забезпечувати високий рівень обслуговування.

Завдяки поєднанню методів кластеризації, теоретико-множинних моделей і системи розподілу є можливість створити інтелектуальне середовище, яке реагує на потреби сучасних застосунків IoT та забезпечує їх надійну роботу. Це дослідження є першим у вивченні розвитку систем туманного обчислення для майбутніх викликів і можливостей.

Список літератури

1. Alam T. Cloud-based IoT applications and their roles in smart cities. *Smart Cities*. Vol. 4(3), 2021. P. 1196–1219. DOI:10.3390/smartcities4030064

2. Al-Haija Q.A. Top-Down Machine Learning-Based Architecture for Cyberattacks Identification and Classification in IoT Communication Networks. *Frontiers in Big Data*, Vol. 4. 2022. P. 1–18. DOI: https://doi.org/10.3389/fdata.2021.782902

3. Uthayakumar J., Vengattaraman T. and Dhavachelvan P. A new lossless neighborhood indexing sequence (NIS) algorithm for data compression in wireless sensor networks. *Ad Hoc Networks*, Vol. 83, 2019. P. 149–157. DOI: https://doi.org/10.1016/j.adhoc.2018.09.009

4. Al-Hawawreh M., Elgendi I. and Munasinghe K. An Online Model to Minimize Energy Consumption of IoT sensors in Smart Cities. *IEEE Sensors Journal*, Vol. 22(20), 2022. P. 19524–19532. DOI: https://doi.org/10.1109/JSEN.2022.3199590

5. W. Jing, G. Chen, and Y. Cheng, DBSCAN-PSM: an improvement method of DBSCAN algorithm on Spark, International Journal of High Performance Computing and Networking, Vol.13, No.4, 417 p., 2019. URL: https://www.inderscience.com/offers.php?id=99265

6. Cordova I., Moh T. DBSCAN on Resilient Distributed Datasets, *International Conference on High Performance Computing Simulation (HPCS). IEEE*. 2015, P. 531–540. DOI: https://doi.org/10.1109/HPCSim.2015.7237086

7. Augustine S., Ananth J.P. Taylor kernel fuzzy C-means clustering algorithm for trust and energy-aware cluster head selection in wireless sensor networks. *Wireless Networks*, Vol. 26(7), 2020. P. 5113–5132. DOI: https://doi.org/10.1007/s11276-020-02352-w

8. He Y., Tan H., Luo W., Mao H., Ma D., Feng S., and Fan J., MR-DBSCAN: An Efficient Parallel Density-Based Clustering Algorithm Using MapReduce, *IEEE 17th International Conference on Parallel and Distributed Systems. IEEE*, 2011. P. 473–480. DOI: 10.1109/ICPADS.2011.83

9. D. Han, A. Agrawal, W.-k. Liao, and A. Choudhary, Parallel DBSCAN Algorithm Using a Data Partitioning Strategy with Spark Implementation, in 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018. P. 305–312. URL: https://www.scholars.northwestern.edu/en/publications/parallel-dbscan-algorithm-using-a-data-partitioning-strategy-with

10. Gong Y., Sinnott R. O., and Rimba P. RT-DBSCAN: RealTime Parallel Clustering of Spatio-Temporal Data Using SparkStreaming, *Computational Science* 2018. P. 524–539. DOI: 10.1007/978-3-319-93698-7_40

11. Ali N., Hamida S, Cherradi B et al. A computational performance study of unsupervised data clustering algorithms on GPU. *2nd international conference on innovative research in applied science, engineering and technology (IRASET). IEEE*, Meknes, 2022. P 1–6. DOI:10.1109/IRASET52964.2022.9737871

12. Cook S. CUDA programming: a developer's guide to parallel computing with GPUs. Elsevier, MK, Amsterdam; Boston. 2013. 591 p. URL: https://usermanual.wiki/Pdf/Shane20CookCUDA20programming20A20developers20guide20to20parallel20computing20with20GPUs Morgan20Kaufmann202012.1739933505/help

13. Fritz F, Schmid M, Mottok J. Accelerating real-time applications with predictable work-stealing. Architecture of computing systems. ARCS. Springer International Publishing, Cham, 2020. P. 241–255. URL: https://europepmc.org/article/pmc/pmc7343420

14. Li Y, Zhao K, Chu X, Liu J. Speeding up k-Means algorithm by GPUs. *Journal of Computer and System Sciences* Vol. 79, Issue 2, 2013. P. 216– 229. DOI: https://doi.org/10.1016/j.jcss.2012.05.004

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) *Innovative technologies and scientific solutions for industries. 2023. No. 3 (25)*

15. Sanders J, Kandrot E. CUDA by example: an introduction to general-purpose GPU programming. Addison-Wesley, Upper Saddle River, NJ. 2013. 311 p. URL: https://edoras.sdsu.edu/~mthomas/docs/cuda/cuda_by_example.book.pdf

16. Wasif M. K., Narayanan P. J. Scalable clustering using multiple GPUs. *18th international conference on high performance computing. IEEE*. Bengaluru, 2011. P. 1–10. DOI: https://doi.org/10.1016/j.engappai.2017.10.023

17. Rodriguez D., Gomez D., Alvarez D., Rivera S. A review of parallel heterogeneous computing algorithms in power systems. *Algorithms.* 2021. Vol. 14(10). 275 p. DOI: https://doi.org/10.3390/a14100275

References

1. Alam, T. (2021), "Cloud-based IoT applications and their roles in smart cities". *Smart Cities*. Vol. 4(3). P. 1196–1219. DOI:10.3390/smartcities4030064

2. Al-Haija, Q.A. (2022),"Top-Down Machine Learning-Based Architecture for Cyberattacks Identification and Classification in IoT Communication Networks". *Frontiers in Big Data*, Vol. 4. P. 1–18. DOI: https://doi.org/10.3389/fdata.2021.782902

3. Uthayakumar, J., Vengattaraman, T. and Dhavachelvan, P. (2019),"A new lossless neighborhood indexing sequence (NIS) algorithm for data compression in wireless sensor networks". *Ad Hoc Networks*, Vol. 83. P. 149–157. DOI: https://doi.org/10.1016/j.adhoc.2018.09.009

4. Al-Hawawreh, M., Elgendi I. and Munasinghe K. (2022), "An Online Model to Minimize Energy Consumption of IoT sensors in Smart Cities". *IEEE Sensors Journal*, Vol. 22(20). P. 19524–19532. DOI: https://doi.org/10.1109/JSEN.2022.3199590

5. Jing, W., Chen, G., and Cheng, Y. "DBSCAN-PSM: an improvement method of DBSCAN algorithm on Spark". International Journal of High Performance Computing and Networking, Vol. 13, No.4, 417 p., 2019. available at: https://www.inderscience.com/offers.php?id=99265

6. Cordova, I., Moh, T. (2015), "DBSCAN on Resilient Distributed Datasets". *International Conference on High Performance Computing Simulation (HPCS). IEEE*. P. 531–540. DOI: https://doi.org/10.1109/HPCSim.2015.7237086

7. Augustine, S., Ananth, J.P. (2020), "Taylor kernel fuzzy C-means clustering algorithm for trust and energy-aware cluster head selection in wireless sensor networks". *Wireless Networks*, Vol. 26(7). P. 5113–5132. DOI: https://doi.org/10.1007/s11276-020-02352-w

8. He, Y., Tan, H., Luo, W., Mao, H., Ma, D., Feng, S., and Fan, J. (2011), "MR-DBSCAN: An Efficient Parallel Density-Based Clustering Algorithm Using MapReduce", *IEEE 17th International Conference on Parallel and Distributed Systems. IEEE*. P. 473–480. DOI: 10.1109/ICPADS.2011.83

9. Han, D., Agrawal, A., Liao, W., and Choudhary, A. "Parallel DBSCAN Algorithm Using a Data Partitioning Strategy with Spark Implementation", in 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018. P. 305–312. available at: https://www.scholars.northwestern.edu/en/publications/parallel-dbscan-algorithm-using-a-data-partitioning-strategy-with

10. Gong, Y., Sinnott, R. O., and Rimba, P. (2018), "RT-DBSCAN: RealTime Parallel Clustering of Spatio-Temporal Data Using SparkStreaming", *Computational Science*. P. 524–539. DOI:10.1007/978-3-319-93698-7_40

11. Ali N., Hamida S, Cherradi B et al. (2022), "A computational performance study of unsupervised data clustering algorithms on GPU". *2nd international conference on innovative research in applied science, engineering and technology (IRASET). IEEE*, Meknes, 2022. P 1–6. DOI:10.1109/IRASET52964.2022.9737871

12. Cook, S. "CUDA programming: a developer's guide to parallel computing with GPUs". Elsevier, MK, Amsterdam; Boston. 2013. 591 p. available at: https://usermanual.wiki/Pdf/Shane20CookCUDA20programming20A20developers20guide20to20parallel20computing20with20GPUs Morgan20Kaufmann202012.1739933505/help

13. Fritz F, Schmid M, Mottok J. "Accelerating real-time applications with predictable work-stealing. Architecture of computing systems". ARCS. Springer International Publishing, Cham, 2020. P. 241–255. available at: https://europepmc.org/article/pmc/pmc7343420

14. Li Y, Zhao K, Chu X, Liu J. (2013), "Speeding up k-Means algorithm by GPUs". *Journal of Computer and System Sciences* Vol. 79, Issue 2. P. 216–229. DOI: https://doi.org/10.1016/j.jcss.2012.05.004

15. Sanders, J, Kandrot, E. "CUDA by example: an introduction to general-purpose GPU programming". Addison-Wesley, Upper Saddle River, NJ. 2013. 311 p. available at: https://edoras.sdsu.edu/~mthomas/docs/cuda/cuda_by_example.book.pdf

16. Wasif, M. K., Narayanan, P. J. (2011), "Scalable clustering using multiple GPUs". *18th international conference on high performance computing. IEEE*. Bengaluru. P. 1–10. DOI: https://doi.org/10.1016/j.engappai.2017.10.023

Сучасний стан наукових досліджень та технологій в промисловості. 2023. № 3 (25)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

17. Rodriguez, D., Gomez, D., Alvarez, D., Rivera, S. (2021), "A review of parallel heterogeneous computing algorithms in power systems". *Algorithms.* 2021. Vol.14(10). 275 p. DOI: https://doi.org/10.3390/a14100275

Надійшла 15.09.2023

Відомості про авторів / About the Authors

Резанов Богдан Михайлович – Національний технічний університет "Харківський політехнічний інститут", аспірант кафедри комп'ютерної інженерії та програмування, Харків, Україна; e-mail: brezanov@gmail.com; ORCID ID: http://orcid.org/0000-0002-4113-8781

Кучук Георгій Анатолійович – доктор технічних наук, професор, Національний технічний університет "Харківський політехнічний інститут", професор кафедри комп'ютерної інженерії та програмування, Харків, Україна; e-mail: kuchuk56@ukr.net; ORCID ID: http://orcid.org/0000-0002-2862-438X

Rezanov Bohdan – National Technical University "Kharkiv Polytechnic Institute", Student at the Department of Computer Engineering and Programming, Kharkiv, Ukraine.

Kuchuk Heorhii – Doctor of Sciences (Engineering), Professor, National Technical University "Kharkiv Polytechnic Institute", Professor at the Department of Computer Engineering and Programming, Kharkiv, Ukraine.

MODEL OF ELEMENTAL DATA FLOW DISTRIBUTION IN THE INTERNET OF THINGS SUPPORTING FOG PLATFORM

The subject of the research is models and methods for optimizing resource and task management in the fog computing environment of the Internet of Things (IoT). The increasing number of connected devices and the volumes of data collected in IoT networks make it essential to improve management systems that ensure the optimal distribution of tasks and resources. Fog computing addresses this challenge by distributing computational tasks closer to data sources and end-users. **The goal** of this work is to enhance the efficiency of fog computing technologies to achieve optimal task and resource allocation in IoT networks. **The main tasks** of this work are as follows. Firstly, considering the diverse requirements of computational resources and tasks in IoT, reviewing existing methods and developments in the field is necessary. Secondly, it is essential to investigate and compare clustering methods, particularly DBSCAN and C-Means, for effective resource management. The DBSCAN clustering method enables efficient task distribution based on their location, while the C-Means method allows grouping resources based on their characteristics. The final task involves developing a mathematical model that considers input parameters such as system response, cluster resource requirements, data proximity to processing, etc. This model will enable the analysis of potential scenarios and decision-making regarding the optimal distribution of tasks and resources in the IoT environment. **Conclusion**. This research aims to solve the urgent problem of managing resources and tasks in the fog IoT environment. A review of existing methods and developments in resource and task management in IoT is conducted. DBSCAN and C-Means clustering methods are compared to determine their effectiveness in resource management. A set-theoretic model is developed that considers various parameters for making optimal decisions on the distribution of tasks and resources. It is established that the use of clustering methods and the developed model help to improve system performance and ensure more efficient use of fog computing resources in the IoT environment.

Keywords: fog computing, IoT, DBSCAN, C-Means, clustering, mathematical modeling.

Бібліографічні описи / Bibliographic descriptions

Резанов Б.М., Кучук Г.А. 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

Rezanov, B., Kuchuk, H. (2023), "Model of elemental data flow distribution in the internet of things supporting fog platform", *Innovative Technologies and Scientific Solutions for Industries*, No. 3 (25), P. 88–97. DOI: https://doi.org/10.30837/ITSSI.2023.25.088